TAMPERE POLYTECHNIC

BUSINESS SCHOOL

Author(s): Daniel Mondesire

Degree Programme(s): International BBA

Title:

Month and year: December 2007

Supervisor: Tomi Nakari **Pages:** 68

ABSTRACT

Flander is a leading software testing company in the mobile and telecom segment. The vast majority of their workload is project-based, hence each business endeavor presents a unique set of circumstances in terms of planning and overall pricing. After analyzing historical data, Flander has concluded that a significant bottleneck exists in the preliminary RFQ development and delivery phase. A disproportionate amount of time is spent on generating preliminary RFQ's. In an effort to reduce lag during this phase, Flander has determined that it must implement a certain level of in-process uniformity and standardization. After careful deliberation, Flander established that a software-based pricing tool would provide the optimal solution for alleviating this bottleneck.

TAMPERE POLYTECHNIC

BUSINESS SCHOOL

FINAL THESIS REPORT

Developing a Pricing Tool for Flander Oy

Daniel Mondesire

Degree Programme in International Business
December 2007
Supervisor: Tomi Nakari

TAMPERE 2007

Table of Contents

1 Introduction

Flander is a leading software testing company in the mobile and telecom segment. The vast majority of their workload is project-based, hence each business endeavor presents a unique set of circumstances in terms of planning and overall pricing. After analyzing historical data, Flander has concluded that a significant bottleneck exists in the preliminary RFQ development and delivery phase. A disproportionate amount of time is spent on generating preliminary RFQ's. In an effort to reduce lag during this phase, Flander has determined that it must implement a certain level of in-process uniformity and standardization. After careful deliberation, Flander established that a software-based pricing tool would provide the optimal solution for alleviating this bottleneck.

The main objective of this project is to develop and implement a preliminary pricing tool in accordance with Flander's business requirements. This will be achieved through the utilization of Flander's historical data, the functionality requirements determined by the organization's management and the existing market conditions.

Successful implementation of the pricing tool should ultimately alleviate the lag which currently exists in the preliminary pricing phase of customer negotiations.

1.1 Client Information

Flander is a leading software testing company in the mobile and telecom segment. Our customers include top-tier mobile handset manufacturers, independent software vendors, mobile operators and service providers. Flander is part of Flander Group. With offices in Finland, Sweden and China, the company employs over 330 professionals.

1.1.1 Flander's Mission Statement and Core Values

Satisfied customers

Customer satisfaction is the core objective at Flander. We invest in strong, long-lasting customer relationships. Our field expertise and deep business knowledge help us to better serve our customers and strengthen these bonds. We stay abreast of current technology and market conditions in order to anticipate the needs and expectations of our customers. Our working methods and processes are transparent for our customers. We continuously monitor and improve our customer satisfaction ratings.

Quality of activity

Flander's process-based quality system has been awarded the ISO 9000 certificate and our software development processes adhere to CMMi level 3. Customer satisfaction of the company's subcontracting projects was rated 4.1 on a scale of one to five in 2005.

Flexibility of activity

We aspire to anticipate the future demands of our customers and are proactive towards them. We act humbly and at the same time purposefully, targeting profit for the customer.

Productivity

We are continuously fine-tuning and improving our work methods, enabling us to increase our customers' profits through cost-reduction. The target of our activity is to accelerate and increase the customer's ability to reliably estimate launch dates for their products and services. Our Quality system, customer knowledge and efficient use of new technology assure competitive rates for our customers.

Committed personnel

Our most important capital is our personnel. We value our employees and focus on their livelihood by emphasizing career-development. We invest in the well-being of our employees through competitive salaries, profit-sharing and a host of other employee benefits.

1.1.2 Key Contributors from Flander

The following key personnel from Flander management were involved with the pricing tool at various phases and to varying degrees:

Tuomas Eerola – Head of Business Development
Mika Heikinheimo – Head of Operations, China
Tommi Holmgren – Project Supervisor
Tommi Kankaanpää – CTO
Jouni Linanmaa – Head of Field Testing Operations

The input and assistance of the aforementioned individuals was invaluable to this project and without their concerted effort, the pricing tool would not have come to fruition.

2 Constituents of a Project

Although a preliminary Request for Quote (RFQ) occurs before the actual project commences, it is tied into the project as a whole. Though it is meant to serve as a rudimentary sales figure for the customer, too much deviation in either direction can have disastrous consequences-ranging from outright rejection by the potential customer if it is too high- to severe financial losses if it is underestimated. In order to understand how the RFQ relates to the bigger-picture of the project as a whole, it is necessary to delve deeper into the various components and characteristics which comprise a project.

In order for a task to qualify as a full-fledged project, it must meet certain criteria. The essential characteristics of a project are:

1. It must be temporary: It must have an explicit beginning and end, whereupon the objective or objectives are met or, if deemed futile, the project is terminated. The usage of the term “temporary” is often misleading, as it may be misconstrued as short-term, when in actuality a project has no minimum or maximum limitations on completion time. “Temporary” in this case refers to situation in which the project duration is finite, irrespective of the length of the project.
2. It must generate unique deliverables in the form of products, services or results: *Products* may be either end items or components of a larger system. *Services* may include improved business functions and documentation stemming from a research project is an example of a *result.*
3. Progressive Elaboration: Can be referred to as the series of “micro-projects”, involved en-route to completion of the final deliverable of the project. It can be likened to the individual stations within an assembly line. Each station adds to the product by performing a specific, essential function which must be completed before advancing to subsequent stations.
4. It must include the human element: Projects must be performed by people.
5. It must include resource constraints: Time, money, manpower, facilities, etc. must be of a limited nature which creates restrictions on the project.
6. It must be planned, executed and controlled: Because project resources are limited, using them effectively and efficiently is of prime importance.

(PMBOK 3.0)

2.1 The Project Life Cycle

The entire span of a project, from conception to completion (or termination) is referred to as the project lifecycle. Many organizations adapt the life cycle to suit their needs, but the basic structure remains the same.

The life cycle is composed of 3 distinct parts or phases: the Initial Phase, the Intermediate phase and the Final phase. Refer to figure 2-1. Transitioning from one phase to the next requires production of a deliverable or deliverables. In most cases, deliverables from a previous phase must be evaluated and accepted before progressing to the next logical stage. Although this concept of logical progression is best-practice, it is not always adhered to. Due to time or other resource constraints, phase transitioning may occur before deliverables from a preceding phase have been accepted. This overlapping of phases is referred to as *fast-tracking* and is typically implemented when it has been deemed that the risks are minimal.

RFQ's generally fall into the Initial phase of the project life cycle which may also include a feasibility study to determine whether a project is worth undertaking or whether the associated risks warrant abandonment.

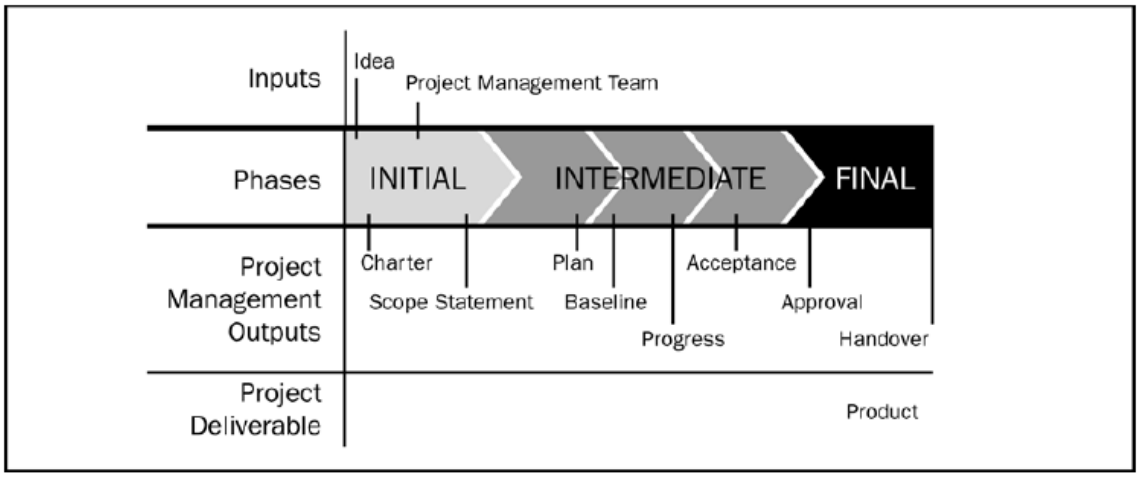


Fig. 2-1 Typical Sequence of Phases in a Project Life Cycle
Source: Project Management Body of Knowledge 3.0

There are several distinct characteristics of the Initial phase which render it a critical aspect of the project life cycle.

This phase bears the highest level of uncertainty, thus imparting it with the highest failure risk of any of the other phases. Shareholders also exhibit their most powerful influence during the Initial phase thereby compounding this inherent instability. An important aspect of this phase, which acts as a counter-balance to the high risk and shareholder influence factors, is the relatively low cost of discovering errors at this stage. Addressing mistakes at this point is the least costly of all other phases. There is a direct relationship between project duration and the cost of resolving faults, whereas an inverse relationship exists between project duration and shareholder influence. Refer to figure 2-2.

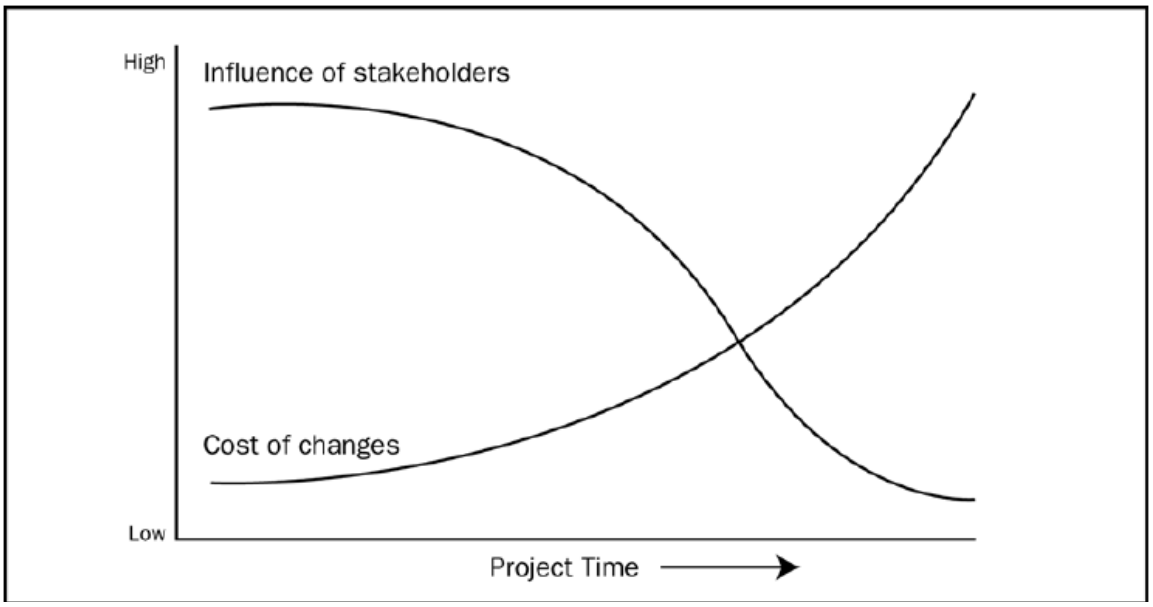


Fig 2-2 Stakeholders' Influence Over Time
Source: Project Management Body of Knowledge 3.0

As previously stated, the Initial stage of a project is the most unstable of all phases. One of an organization's focal points should therefore involve mitigation of these inherently high risks. Diminishing risks may be accomplished in several ways. A sound, well-researched feasibility study, undertaken as a separate entity from the potential project, will provide an accurate barometer of the organization's readiness to undertake a project. The clarity gained from such a study plays a role in garnering shareholder support, creating cohesion and fostering solidarity between management and the stakeholders.

Rapid, accurate and timely communication, particularly in this phase, also serves to partially alleviate the burden of high risk. The pricing tool was specifically designed to facilitate the process of communicating efficiently, both within the organization and on an inter-organizational level.

The major phases of the project life cycle have been briefly touched upon in order to give a rough framework of the process, thus providing a reference point for the pricing tool. The link between inherent high risks in the initial phase and the need to mitigate these risks has been emphasized as a means to establish justification for the creation of the pricing tool.

The next section will further justify the development and implementation of the pricing tool from the standpoint of cost management. It will introduce the concept of managing costs and demonstrate how project-based cost management principles and practices shaped the core elements of the pricing tool.

2.2 Project Cost Management

As a function of the Project Management process, Cost Management covers the gamut, from planning and estimating, to budgeting and controlling costs. The overall objective of this branch of Project Management is to ensure that the project is completed within the approved budget. Cost management is subdivided into the following 3 processes:

1. **Cost Estimating** – Approximating the cost of resources needed to complete a project.
2. **Cost Budgeting** – Aggregating the estimated costs of individual activities or work packages to establish a cost baseline.
3. **Cost Control** – Isolating and influencing the factors that create cost variances and controlling changes to the project budget.

Although these sub-processes are more interdependent upon one another than they are independent of each other, for the purpose and scope of this paper, the focus will be solely on Cost Estimation, while both Cost Budgeting and Cost Control will be excluded.

2.2.1 Cost Estimation Inputs

Cost estimation inputs fall under 2 categories: **Enterprise Environmental Factors** and **Organizational Process Assets**

Enterprise Environmental Factors include:

- **Marketplace conditions**: Which products and services are currently available in the marketplace. Who are the organizations offering these products or services and what are the terms and conditions.
- **Commercial Databases:** Current cost rate of resources through commercial databases which track: human resource costs, skills in addition to standard material and equipment costs.
- **Various internal/external factors**: A multitude of other influences can come into play and may be classified under this subheading. These include: Organizational culture and structure, industry standards such as regulations and quality, product and workmanship standards, existing human resources (accumulated skills and knowledge such as development, design, purchasing and contracting skills), personnel administration (employee performance reviews, training records, hiring and firing policies), infrastructure, stakeholder risk tolerances and finally, project management information systems (configuration management systems, information distribution systems, scheduling software, web interfaces or other automated systems).

Organizational Process Assets include formal and informal cost estimating policies and guidelines such as: business needs (training, market demand, legal requirements or advancements

in technology), Strategic plans (projects should coincide with an organization's strategic objectives) and if necessary, a *product scope description* (documentation of product requirements including specification of products or services which should be rendered as a final outcome of the project undertaken) .These can be sub-categorized as follows:

- **Cost Estimating policies**: If an organization has a set policy, it will adhere to a predefined cost-estimating approach and operate within those boundaries.
- **Cost Estimating templates**: Project teams may have already developed cost estimation templates. These templates will be continuously improved as they are applied with more frequency in real-world situations.
- **Historical Data**: Information pertaining to the project's product or service, which is obtained from various sources within the organization, can influence project cost.
- **Project Files**: Several key players involved in the project may maintain records of prior project performance which are detailed enough to assist in developing cost estimates. In certain areas, individual team members may be responsible for maintaining such records.

(Adapted from PMBOK 3.0)

Flander applied several of the aforementioned Cost Estimation techniques and policies to generate the baseline figures contained within the calculation engine of pricing tool. The inputs used by Flander to produce these figures include: the use of cost estimating templates, the application of set, yet flexible, cost estimation policies, analyses of marketplace conditions, the incorporation of historical data and the utilization of pre-existing project files and commercial databases.

Refer to *Appendix A: Pricing Tool Baseline Figures* for the exact figures. Their extensive project based testing experience supplied ample historical data with which to work and afforded a greater degree of accuracy to the pricing tool. Refer to figure 2-3 for Cost Estimation inputs.

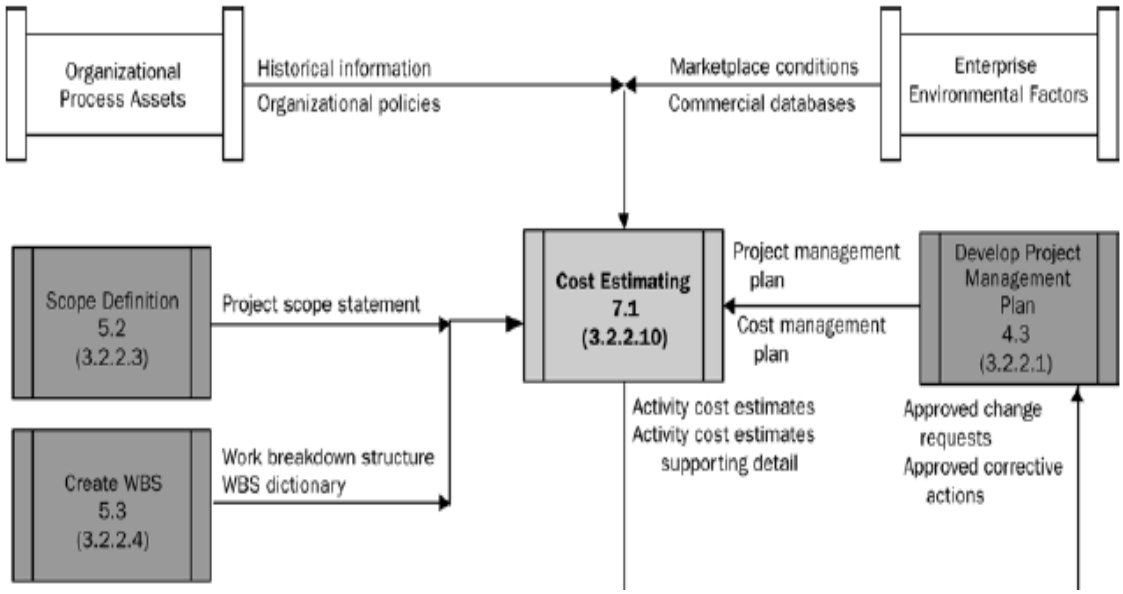


Figure 2-3
Source: Project Management Body of Knowledge 3.0

By its very nature, Cost Estimation is not extremely accurate, hence the terminology "estimation", however accuracy tends to increase (up to a certain point) in direct proportion to the quantity and quality of historical data available. Accuracy issues in RFQ estimation can be attributed to its proximity to project completion:

Cost estimates can benefit from refinement during the course of the project to reflect the additional detail available. The accuracy of a project estimate will increase as the project progresses through the project life cycle.
(PMBOK 3.0)

Now that the source of the raw numerical data at the core of the pricing tool has been analyzed, the next logical step involves the organization of this data as it relates to the tool. This compiling of data in a logical order will be accomplished through the construction and utilization of a database.

The next section will provide a theoretical background behind database design in order to provide a better understanding of the concepts before delving into the specifics of the actual creation of the database.

3 Theoretical Foundations of the Pricing Tool

Section 2.2 established justification for the pricing tool, while section 2.2.1 revealed how the pricing tool uses figures derived from Cost Estimation techniques to drive its calculation engine. This section will provide a basic theoretical aspect of Database design, the core discipline behind the functionality of the pricing tool.

3.1 A Theoretical Approach to Database Design

The pricing tool is nothing more than a relational database, using specific SQL statements, Macros and VBA's wrapped in a UI, as a means to retrieve information. To provide a better understanding of the logic and thought-processes behind the creation of the pricing tool, it is necessary to cover some theoretical aspects of database design.

A database is a collection of data, typically describing the activities of one or more related organizations. Databases usually contain some form of the following elements:

- Entities such as people, departments, salaries etc.
- Attributes or characteristics which describe these entities
- Relationships between these entities

(Ramakrishnan & Gehrke 2001)

Most databases are *relational* in nature. In mathematical terms, *relational* literally means "table based", which accurately describes the nature of these databases. Data is stored in individual tables which are then linked by a series of relationships that are defined during the conceptual modeling phase of database design.

The initial stage of database design involves constructing the conceptual model of the database. This critical phase produces the blueprint from which the database will be built, hence any design flaws will ultimately manifest themselves in the physical construct of the database. It is therefore of prime importance that great care be taken at this juncture to avoid design flaws.

The conceptual model serves several key functions:

- It explicitly describes the information needs
- It facilitates discussion
- It provides clarity which helps prevent mistakes and misunderstandings
- It forms important reference documentation depicting database architecture and describing how an ideal system should behave

(Speelpenning, 2001)

An important sub-phase of conceptual modeling is entity-relationship modeling or ER modeling. The main objective of this stage is to define, exhaustively, all the informational requirements of the database. ER modeling ensures:

- All pieces of information that are required to run a business properly are recognized. Models should be complete. Requirements should be known before you start implementing. Dependencies must be clear.
- Every single piece of required information appears only once in the model. This is an important goal. As soon as a system stores particular information twice, you run into the possibility that this information is not the same in both places. If you are a user of an information system and discover inconsistencies in the data, which information would you to trust? This goal implies that an ideal system does not contain derivable information.
- In the future system, the information is made available in a predictable, logical place; related information is kept together.
- A proper Entity Relationship model leads to a set of logically coherent tables.

(Speelpenning, 2001)

Once the blueprint is drawn-up and checked extensively, construction of the database may commence.

Section 2 delved into the theoretical aspect of project management, then harnessed portions of the theory and applied them in a practical sense to demonstrate the processes implemented by Flander to arrive at the raw numbers used in the calculation engine of the pricing tool.

Section 3 covered some of the basic theoretical aspects of database design in an attempt to provide a solid foundation on which to introduce the concrete process of constructing a database, which is documented in the third section.

Section 4 will cover the development of the pricing tool, in its entirety (albeit condensed and distilled down to the major elements and milestones) from concept to end-product, using sections 2 and 3 as its theoretical foundation.

4 The Pricing Tool: From Concept to Concrete Product

This section will shed light on how the pricing tool was assembled, including concept development, the process of organizing the data into a logical coherent system and determining the most efficient way to arrange the data in order to optimize retrieval accuracy while simultaneously ensuring intuitive, hassle-free updating of the system data.

4.1 The Concept

As per the abstract section of this paper, Flander identified the need for a pricing tool as a means for addressing slow preliminary RFQ response times.

The basic framework of the pricing tool was initially conceived, verbally, through a one-on-one meeting with Mr. Eerola. The premise was explained, and a rough sketch of how the tool should be implemented and what functionality should be incorporated was devised.

The outcome of this meeting was that researching of concepts covered in our initial meeting should commence and those findings be documented in logical fashion to begin “fleshing-out” the pricing tool.

Appendix B: The initial pricing tool proposal, was borne out of this meeting.

The initial proposal was created as a template based on the structure and features of a Symbian-based smartphone. The smartphone was chosen as the platform of the pricing tool as it is an all-inclusive device, therefore a pricing tool derived from a model of its functionality would not only be backward compatible with older devices, but also open and forward-thinking, enabling it to consider existing technology, in addition to accommodating future technological advances.

A smartphone is commonly referred to as an “open phone” because its hardware layout and operating system allow for features to be added rapidly and efficiently as the market develops:

“It’s about the capacity for new features to be added quickly, taking advantage of the ideas of a huge ecosystem of inventors, innovators, and entrepreneurs.” (Wood, 2005)

This justification stems from 4 key smartphone-related trends identified by David Wood:

1. The first trend is that software is becoming ubiquitous. Inanimate objects all around us are becoming smarter and smarter. High-powered software, once found only in stationary computers, now flourishes inside all manner of mobile devices

2. Second, levels of communication keep on rising. More and more types of message are being sent between more and more people and between more and more devices. Mobile

smart devices are constantly joining dynamic networks that potentially make their users into even smarter individuals

3. Third, users are increasingly demanding an external simplicity in the devices and tools they use (even though these devices abound with inner complexity). Users have lost the patience to bother with tedious operating manuals. Users want power, but they want it easy

4. And fourth, users are demanding additional abilities to customize, personalize, and adapt the devices and the tools that they use. They want to be creators and innovators, not just consumers. They want their devices to be unique and distinctive. This brings programmability to the fore. It's where software becomes really soft. It's where smartphones become really personal.

(Wood, 2005)

Rapid market growth for smartphones (100% increase from 2002-2005) and their near guaranteed continued phenomenal growth rate, were other driving forces behind modeling the pricing tool from a smartphone template.

The assurance that the market will continue to demand smartphone technology at such high levels is based on observations in 3 key areas: market evolution, network dynamics and technological progress. This can be further subdivided into 7 critical factors:

1. Moore's Law means that, for the same cost, more and more powerful hardware can be supplied; tomorrow's smartphones will have as much computing power as yesterday's PCs

2. New generations of phone networks (3G, 3.5G, 4G, and so on) will allow the speedy transmission of ever larger amounts of data, both satisfying and whetting still more user demand

3. More powerful devices and more powerful networks jointly enable the provision of attractive add-on services, created by third parties, which in turn increase the market pull for devices capable of supporting such services

4. The cumulative operation of software means that new services and applications can piggy-back on the functionality and power of previous services and applications, with striking, innovative results

5. Many of these services are community-oriented: the more people who take part in these services, the more valuable these services become (this is sometimes called Metcalfe's Law)

6. As people discover the benefits of mobile online gaming, mobile commerce, and so on, they will spread this message by word-of-mouth, so that the communities of smartphone users swell in size

7. Phone network operators have a strong interest in ensuring that phone users are attracted to make regular use of services that involve greater amount of data transfer (and which therefore attract higher fees).

(Wood, 2005)

These key factors are referred to as the "virtuous cycle", a process in which these trends feed each other harmoniously and continue to spur growth. Refer to figure 4-1.

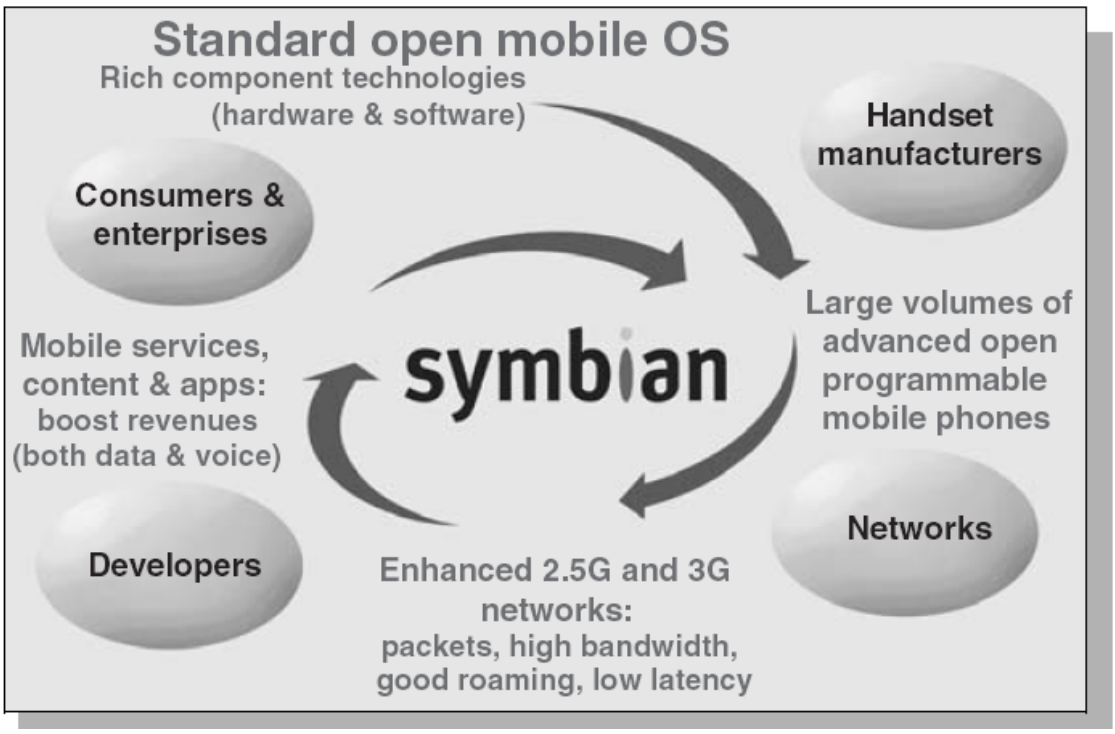


Fig. 4-1 The Virtuous Cycle
Source: Symbian for Software Leaders

4.1.1 Proposal Fine-tuning: Version 2.0

Once more extensive research into the topics covered in section 4.1 was completed. A meeting was conducted in order to formally present the findings and critique the results.

The general consensus was that the initial proposal was, as expected at such an early phase, incomplete, yet moving in the right direction. After careful deliberation, the following feedback was generated:

- More phone-based options should be added. This included both hardware and software related features
- A more comprehensive and customizable language selection list should be implemented
- A means of tracking customer information and logging user interaction should be incorporated
- Openness and flexibility must be worked into the tool
- Adding these features would require a shift from a simple form-based, check-list style tool, to a full-featured, deeply customizable pricing-configurator style tool

The final pricing tool proposal was subsequently drafted, taking into consideration these improvement suggestions. For Reference, *Appendix C: Pricing Tool Final Proposal.*

4.2 Transforming the Numbers: Data from a Visual Perspective

As denoted in section 2.2.1 the numerical data for the pricing tool's calculation engine was originally documented in a series of Excel sheets. Although the data was well-organized, explicit and thorough, it was difficult in some instances to see the numerical relationships clearly.

In order to expedite the process of extracting the data from the Excel sheets, compiling the numbers into a database and specifying the correct relationships between them, a reinterpretation of the data was required.

Using MS Visio, the Excel sheet values were translated into visual form in order to gain a better perspective of how the variables relate to each other.

Through the process of color-coding and stacking, clarity was achieved. These techniques provided a more clear perspective on how related variables should be grouped. The visual tool enabled tracing of multiple relationships to a single convergence point. This allowed for strategic placement of sets of multipliers so that one multiplier variable could be manipulated to influence all members in the associated group.

Each of the testing package proposals from the Excel sheet was mapped-out as an individual diagram and combined into a 3-sheet Visio document.

The visual reference to the pricing packages was an invaluable tool for depicting some of the complex relationships and accurately porting them into the MS Access database.

Refer to *Appendix D: All Levels* for the graphical depiction of the Excel sheet variables.

5 Constructing the Database: An Overview

The final phase was the construction of the database. This stage consisted of the porting of information from the Excel documents to an Access database and utilizing documentation derived from preceding phases, such as the final proposal and the Visio sheets to accurately construct all the relevant relationships.

The database behind the pricing tool is composed of over 70 separate elements, from tables and queries to forms and reports. Linking these elements together is a web of SQL statements, Macros, VBA's and other custom code. To elucidate in even minor detail on this sheer volume of material would be long-winded and well beyond the technical scope of this paper. Therefore, only the main aspects of the tool will be covered, including the thought processes involved and where necessary, provide brief explanations of some of the functionality.

The tool was developed incorporating many of the theoretical fundamentals of database design outlined in section 2, using Microsoft Access. MS Access was chosen because of its support for SQL, Macros and VBA's and because it was easily accessible for the members of the target user group.

The User Interface layer has been fully customized to Flander standards and is easy to upgrade or further customize on demand.

The majority of the user input is converted into SQL statements and presented to the database in the form of on-the-fly queries although some input is routed to a series of stored queries. These stored queries are used to handle the most commonly requested information from the database and because the results are not actually assembled on-the-fly, it is less taxing on the database and is therefore a form of optimization.

5.1 Pricing Tool Screen Break-down

The pricing tool is composed of 5 main screens and 3 sub-screens for a total of 8 main interactive screens. Each of these 8 components serves a unique purpose and follows a logical which builds off subsequent screens. The theory and logic involved in developing each screen is will now be covered.

5.1.1 Screen 1: Customer Information

The *Customer Information* portion represents the initial screen of the pricing tool's user interface. It records relevant information such as the user and customer and creates a tag which will be

attached to all subsequent database transactions performed within that particular session. This is critical for accurately tracking and monitoring database activity.

The customer information screen is the graphical representation of the database table of the same name. Data input into the fields of this form are stored in their corresponding fields on the table. Once the desired data is input, the user proceeds to screen 2 by clicking the “Next” button at the bottom of the form.

Each click of the “Next” button records the field data and inserts it as a new record within the table. Once the data is stored in the table, it can no longer be edited from the form view.

Only users with administrator rights may access and edit stored table data. This policy adheres to both the *Secrecy and Integrity* aspects of the 3 fundamental security objectives of database design:

1. **Secrecy:** Information should not be disclosed to unauthorized viewers
2. **Integrity:** Only authorized users should be allowed to modify data
3. **Availability:** Authorized users should not be denied access

(Ramakrishnan&Gehrke 2001)

Clicking the “Next” button anchored to this form performs multiple functions, or “On Event” commands, by executing numerous lines of custom-written code. The code behind this button looks like this:

```
Private Sub CustID_Submit_Click()
On Error GoTo Err_CustID_Submit_Click


DoCmd.GoToRecord , , acNewRec
DoCmd.OpenForm “frm2Testing_Profile”, acViewNormal

Exit_CustID_Submit_Click:
Exit Sub

Err_CustID_Submit_Click:
MsgBox Err.Description
Resume Exit_CustID_Submit_Click

End Sub
Private Sub LaunchForm2_Click()
On Error GoTo Err_LaunchForm2_Click

Dim stDocName As String
Dim stLinkCriteria As String
```

stDocName = “Form2_Testing_Profile”
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_LaunchForm2_Click:
Exit Sub

Err_LaunchForm2_Click:
MsgBox Err.Description
Resume Exit_LaunchForm2_Click

If IsNull(Me.Date) Then
Me.cmbDay.Value = Day(Date)
Me.cmbMonth.Value = Month(Date)
Me.cmbYear.Value = Year(Date)
Else
Me.cboDay = Day(Me.Date.Value)
Me.cboMonth = Month(Me.Month.Value)
Me.cboYear = Year(Me.Year.Value)
End If
End Sub

This code performs the following actions:

1. Inserts data from the form fields into the corresponding fields of the table
2. Closes *frm1Customer_Information*
3. Opens the object *frmPackageCriteriaUpdate* in form view with all fields set to null

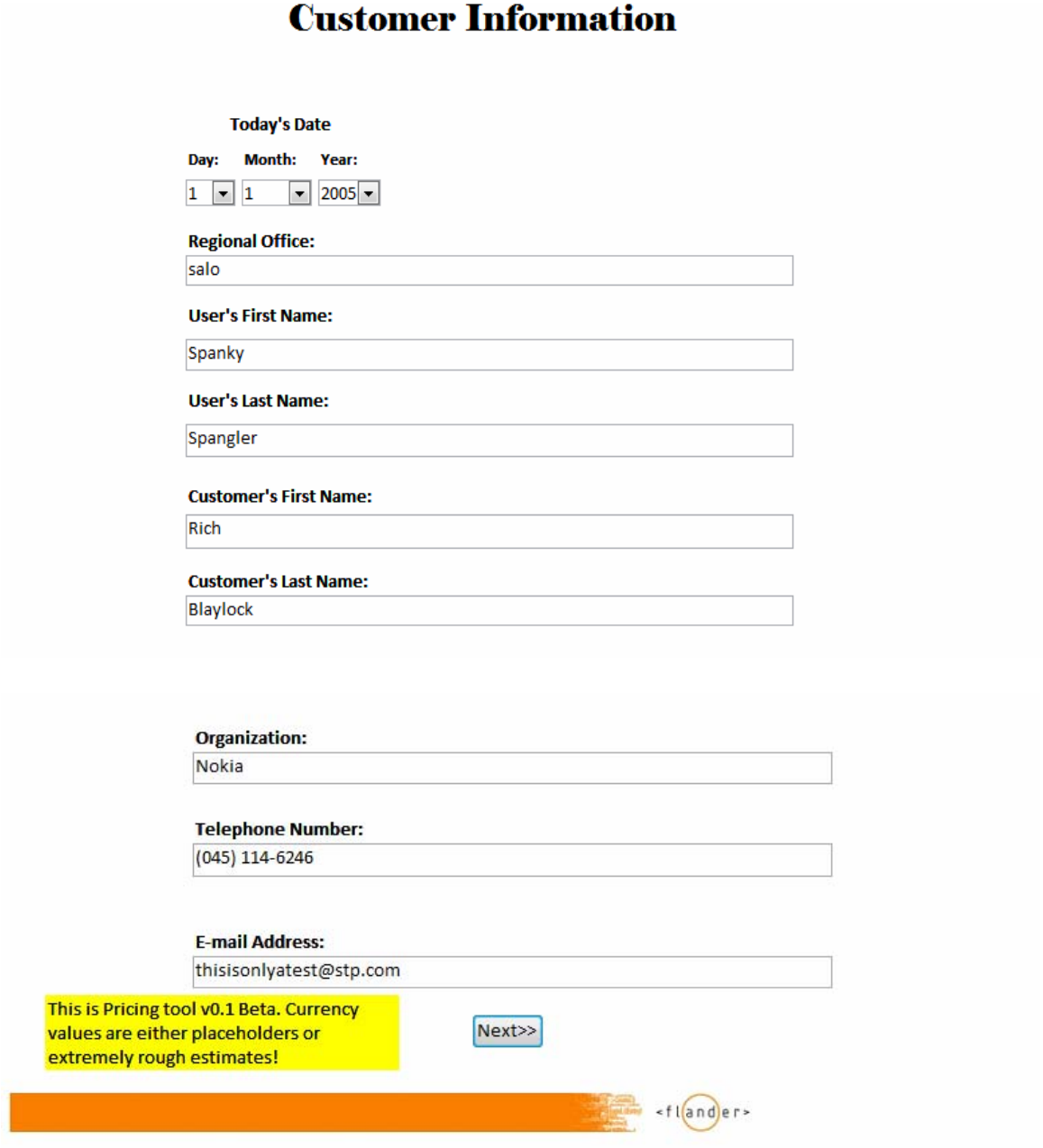


Figure 5-1: Pricing Tool - Screen 1

5.1.2 Screen 2: Pricing Criteria

After the data input in the initial screen is successfully recorded to the table by clicking the "Next" button, screen 1 is closed and the user is presented with screen 2, *Pricing Criteria*, shown as a screenshot below:

Figure 5-2: Screen 2 - Pricing Criteria

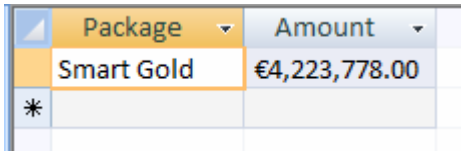


Figure 5-2a: "Recommend Package" results from Screen 2

This screen presents the user with 3 combo-boxes requesting information about the sales target, the depth and the extent of language testing respectively.

Once the appropriate variables are selected, the user initiates further action by clicking on the "Recommend Package" button which executes a multi-functional batch of code.

As dictated by the code, the variables entered into these combo-boxes are converted into SQL statements, generating a Dynamic ("on the fly") query, *qryCriteria*, which in turn retrieves data from table *tblCmbParam.*

A dynamic query was utilized in this phase rather than a static query for 2 important reasons:

1. Static queries require the data to be pre-compiled and stored. This data may only be needed for a few specific queries, but must be stored nonetheless. In addition to having to house this extra data, additional code must be written to access it. This unnecessary storage of data and it's corresponding code is referred to as *code bloat* and after it reaches a certain level, can decrease the performance of information retrieval within a database
2. All things equal, dynamic query results are generated twice as fast as static results. This is due to the fact that dynamic queries are more specialized and intuitive whereas static queries or stored procedures follow the same path to obtain results, even if the execution plan is slower (Bouma 2007)

The query results are recombined in the UI to generate the output, "Recommend Package", based on the specific variables selected. For a simplified view of this process, refer to Figure 5-3 below.

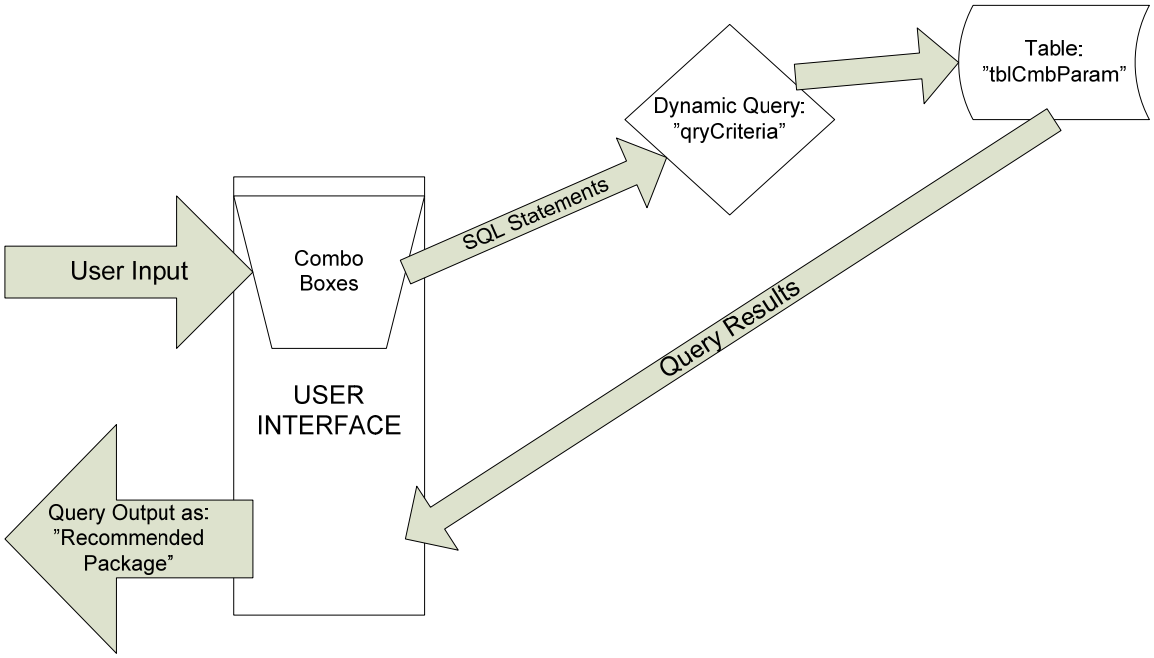


Figure 5-3: Simplified Workflow of Screen 2

The code responsible for controlling the functionality within this screen of the UI, resulting in the workflow shown above looks like this:

```
Option Compare Database


Private Sub cmbObj_AfterUpdate()

End Sub

Private Sub Command6_Click()
Dim db As DAO.Database
Dim qdf As DAO.QueryDef
Dim strSQL As String
Set db = CurrentDb()
Set qdf = db.QueryDefs("qryCriteria")
strSQL = "SELECT tblCmbParam.Package, tblCmbParam.Amount " & _
"FROM tblCmbParam " & _
"WHERE tblCmbParam.Objective='" & Me.cmbObj.Value & "' " & _
"AND tblCmbParam.LangAmnt='" & Me.cmbLang.Value & "' " & _
"AND tblCmbParam.Depth='" & Me.cmbTcover.Value & "' " & _
"ORDER BY tblCmbParam.Amount;"
qdf.SQL = strSQL
DoCmd.OpenQuery "qryCriteria"
DoCmd.Echo False
If Application.SysCmd(acSysCmdGetObjectState, acQuery, "qryCriteria") = acObjStateOpen
Then
DoCmd.Close acQuery, "qryCriteria"
End If
qdf.SQL = strSQL
DoCmd.OpenQuery "qryCriteria"
DoCmd.Echo True
Set qdf = Nothing
Set db = Nothing
End Sub
```

The code is anchored to the "Recommend Package" button and is executed as an "After Update" event procedure. It performs the following actions in sequence:

1. Identifies the table which will be the source of information for the dynamic query
2. Checks the table for exact string matches based on the query variables
3. If an exact match is found, it displays the values of the “Package” and “Amount” fields within the matching record. If no match is found, the resulting query is blank

At this point, via the on-screen navigation buttons, the user may choose to either go back to the previous screen, or progress to the subsequent, database form, Screen 3.

5.1.3 Screen 3: Pricing Criteria Part 3

The third screen is a continuation of the pricing criteria selection process which commenced on the previous screen. It was implemented as a means to fine-tune user response to question 2 in screen 2: the amount of languages to be tested.

The UI for this screen is composed of 2 major sub-sections. The first section is a three-part cascading series of combo-boxes using real-time filtering and enabling multi-selection.

The second subsection poses 3 questions, critical in price determination, and essential for establishing the necessity of providing the user the sixth and final screen.

Refer to Figure…. below for a screenshot of this sub-section of the UI.

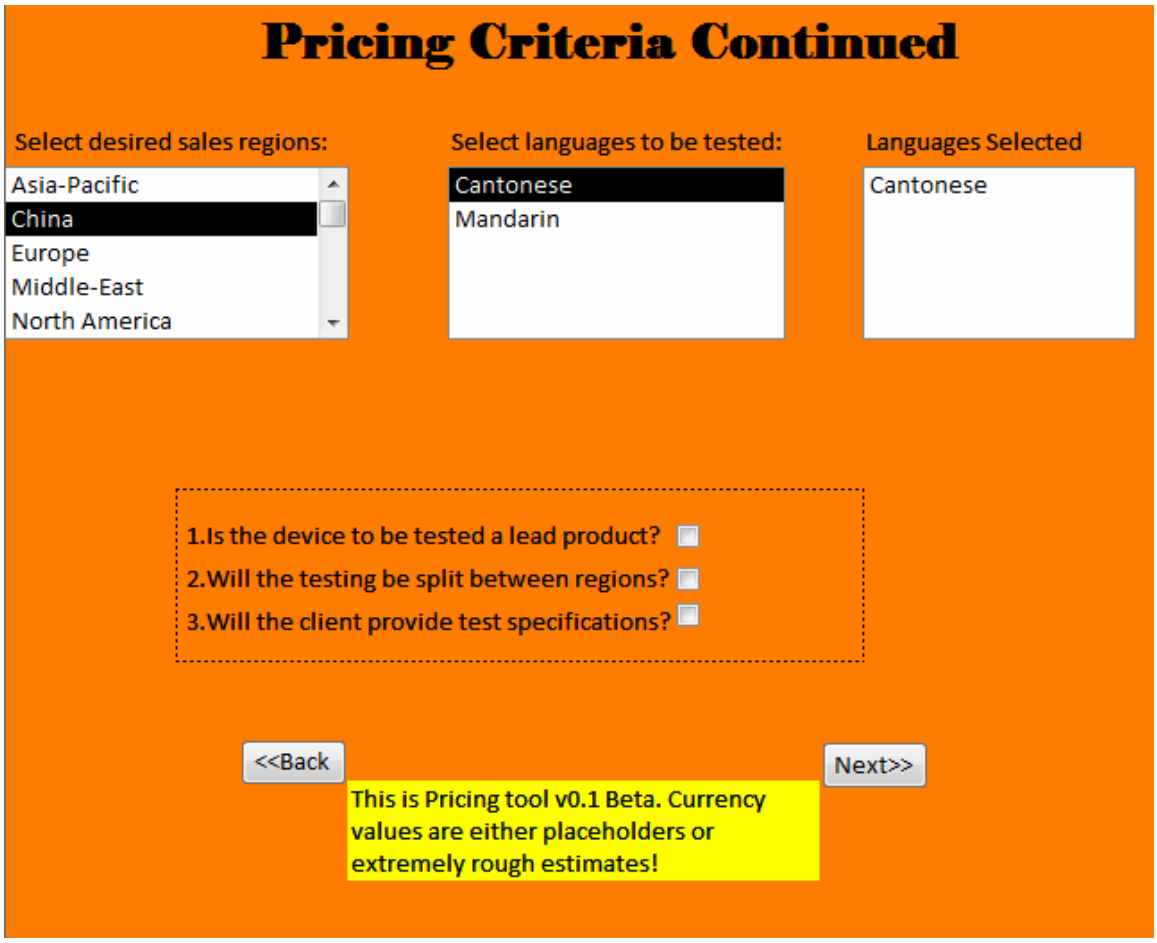


Figure 5-4: Screen 3 - Pricing Criteria Part 2

The cascading combo-boxes utilize a complex series of code which pass the input from the first box (in the form of SQL statements) through a filter to obtain the results displayed in combo-box 2. The results of the first to boxes are then tallied and stored in the third box.

This smooth relaying of information between the 3 combo-boxes is achieved through the use of several dynamic queries. The code behind this is as such:

Private Sub Command6_Click()

End Sub

Private Sub Add_Click()
End Sub

Private Sub lstLangs_AfterUpdate()
Dim db As DAO.Database

```
Dim qdf As DAO.QueryDef
Dim varItem As Variant
Dim strCriteria As String
Dim strSQL As String
Set db = CurrentDb()
Set qdf = db.QueryDefs("qrySeleLangs")
For Each varItem In Me!lstLangs.ItemsSelected
strCriteria = strCriteria & ",'" & Me!lstLangs.ItemData(varItem) & "'"
Next varItem
If Len(strCriteria) = 0 Then
MsgBox "You did not select anything from the list" _
, vbExclamation, "Nothing to find!"
Exit Sub
End If
strCriteria = Right(strCriteria, Len(strCriteria) – 1)
strSQL = "SELECT Languages FROM qryLangbyReg " & _
"WHERE qryLangbyReg.Languages IN(" & strCriteria & ");"
qdf.SQL = strSQL
Me.Requery
Me.Refresh
Set db = Nothing
Set qdf = Nothing
End Sub

Private Sub lstReg_AfterUpdate()
Dim db As DAO.Database
Dim qdf As DAO.QueryDef
Dim varItem As Variant
Dim strCriteria As String
Dim strSQL As String
Set db = CurrentDb()
Set qdf = db.QueryDefs("qryLangbyReg")
For Each varItem In Me!lstReg.ItemsSelected
strCriteria = strCriteria & ",'" & Me!lstReg.ItemData(varItem) & "'"
Next varItem
If Len(strCriteria) = 0 Then
MsgBox "You did not select anything from the list" _
, vbExclamation, "Nothing to find!"
Exit Sub
End If
strCriteria = Right(strCriteria, Len(strCriteria) – 1)
strSQL = "SELECT Languages FROM tblDevBackgroundInfo " & _
"WHERE tblDevBackgroundInfo.Regions IN(" & strCriteria & ");"
qdf.SQL = strSQL
Me.Refresh
```

Set db = Nothing
Set qdf = Nothing

End Sub
Private Sub Command24_Click()
On Error GoTo Err_Command24_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "frm3Bronze_Subtotals"
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command24_Click:
Exit Sub

Err_Command24_Click:
MsgBox Err.Description
Resume Exit_Command24_Click

End Sub
Private Sub Command25_Click()
On Error GoTo Err_Command25_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "frmPackageCriteriaUpdate"
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command25_Click:
Exit Sub

Err_Command25_Click:
MsgBox Err.Description
Resume Exit_Command25_Click

End Sub
Private Sub Command26_Click()
On Error GoTo Err_Command26_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = "frm2Testing_Profile"

DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command26_Click:
Exit Sub

Err_Command26_Click:
MsgBox Err.Description
Resume Exit_Command26_Click

End Sub

The code performs several functions in series. These functions include:

1. Specifying which table to use for the dynamic queries
2. Executing the first dynamic query and updating the second combo-box using an automatic refresh command
3. Executing the second dynamic query based on the filtered results of the initial query and updating the third combo-box using the auto-refresh command

The 3 questions making up the second section of this screen were specifically chosen because of their strong ramifications in influencing project pricing.

When dealing with complex devices such as smartphones, the development time of a lead-product (one incorporating some feature which has no predecessor), including testing, is expected to take longer than 9 months, whereas a project which does not involve a lead-product is highly likely to be completed in under 9 months (Wood 2005).

Each month ahead of or behind schedule significantly affects project costs which, directly impact pricing, making this a critical question to incorporate within a tool aimed at accurately estimating project costs.

The issue of splitting the testing workload between regions is particularly important as Flander conducts major operations in China. Depending on the nature of the testing project, Flander may be able to translate their position in China into cost savings which can be directly passed down to the customer.

The third question, determining who will provide the test specification, is another element crucial to accurately pricing a project. Creating product specifications require large amounts of man-power and working hours.

The product specification is "…a sizable document, containing the accumulated output of several scores of person-years of writing effort." (Wood 2005)

The cost of compiling this documentation must therefore be factored into the calculation engine of any tool which attempts to estimate the cost of a testing project.

5.1.4 Screen 4: Pricing Criteria Part 3

Unless region splitting was chosen in the previous phase, screen four represents the final checklist of pricing criteria utilized by the tool.

The cost estimation metrics used here are derived from product-specific features. There are 2 main sections of this screen:

1. Selection of Phone type
2. Selection of individual features

Refer to Figure 5-5 below for reference.

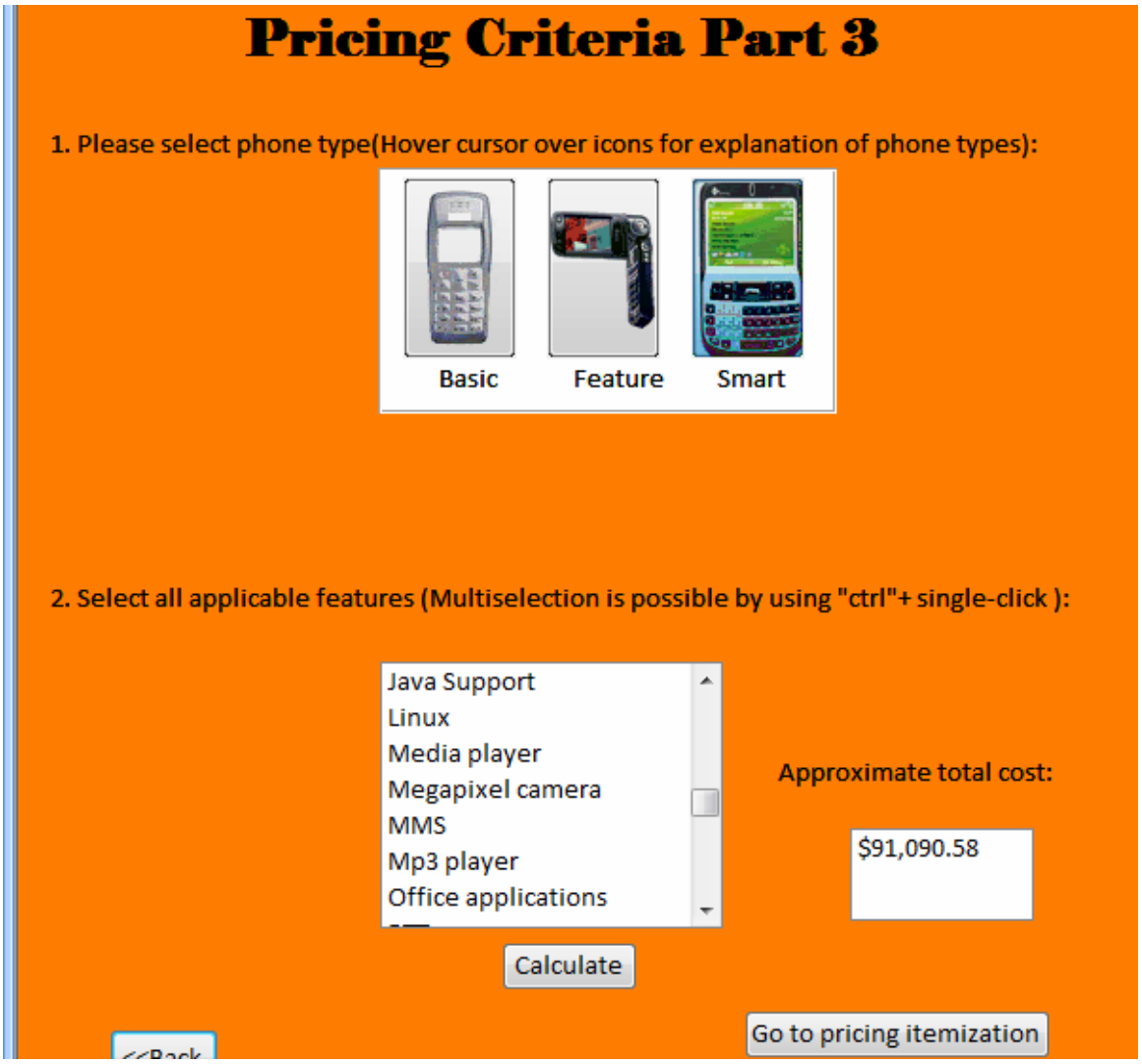


Figure 5-5: Screen 4 - Pricing Criteria Part 3

The phone type selection option is meant to serve as a very rough filter, placing the user closer to an estimate based on the average costs of testing a phone which falls into the specified category.

After choosing the phone type, fine-tuning of the pricing estimate is enabled by allowing for the inclusion or exclusion of specific features. Multiple-selection of features is possible and the total approximate cost is updated each time the “Calculate” button is pressed.

Manipulating price based on feature incorporation is justified by the inherent need to test some features more thoroughly than others, such features, would thus have more impact on pricing estimation than their less testing-intensive counterparts.

Wood groups one particular set of features with these characteristics under the same umbrella. These features require a specialized set of test cases referred to as IOT, or interoperability testing. Features such as Bluetooth, infrared and USB connectivity fall under this category (Wood 2005)

This screen uses a custom-code similar to the one used in the cascading combo-boxes in Screen 3. When the "Calculate" button is activated, selections made in the list-box are passed to a dynamic query which derives its data from the *Phone_Features* table. The package cost field for each feature selected is then summed on-the-fly and the output is passed to the "Approximate total Cost" list-box.

Here is the code behind the functionality of the "Calculate" button in Screen 4:

```
Private Sub Detail_Click()

End Sub


Private Sub PfeatLbox_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub phoneFeatsub_Click()
Dim db As DAO.Database
Dim qdf As DAO.QueryDef
Dim varItem As Variant
Dim strCriteria As String
Dim strSQL As String
Dim strPhonetype As String
Set db = CurrentDb()
Set qdf = db.QueryDefs("MultiSelect")
If Len(strPhonetype) = 1 Then
strCriteria = Right(strCriteria, Len(strCriteria) – 1)
strSQL = "SELECT Sum(SilverCost) FROM Phone_features " & _
"WHERE Phone_features.Features IN(" & strCriteria & ");"
qdf.SQL = strSQL
RunCommand acCmdRefresh
End If

If Len(strPhonetype) = 2 Then
strCriteria = Right(strCriteria, Len(strCriteria) – 1)
strSQL = "SELECT Sum(GoldCost) FROM Phone_features " & _
"WHERE Phone_features.Features IN(" & strCriteria & ");"
qdf.SQL = strSQL
RunCommand acCmdRefresh
End If
```

The code performs the following actions in sequence:

1. Specifies the table from which the data will be selected
2. Runs an automatic refresh operation after each list-box selection and collects the input as string data, building a dynamic query called *strSQL*
3. Sums the package pricing columns from the table *Phone_Features* based on the *strSQL* variables and displays the results in a list-box labeled *list112*

The navigation buttons allow the user to back-track to the previous screen or progress to the next screen in logical order, via the “Go to pricing itemization” button.

5.1.5 Screen 5: Pricing Itemization

The pricing itemization screen represents the most diverse link in the pricing tool. This junction branches off into 3 separate paths, in addition to incorporating the ability to back-track to the previous screen.

At this stage, the user is presented with 3 alternative flows, enabling navigation via one of the following paths:

1. Screen 5A: Custom Multipliers
2. Screen 5B: Itemization by Sub-package
3. Screen 5C: Splitting testing Between Regions

Refer to the screenshot below:

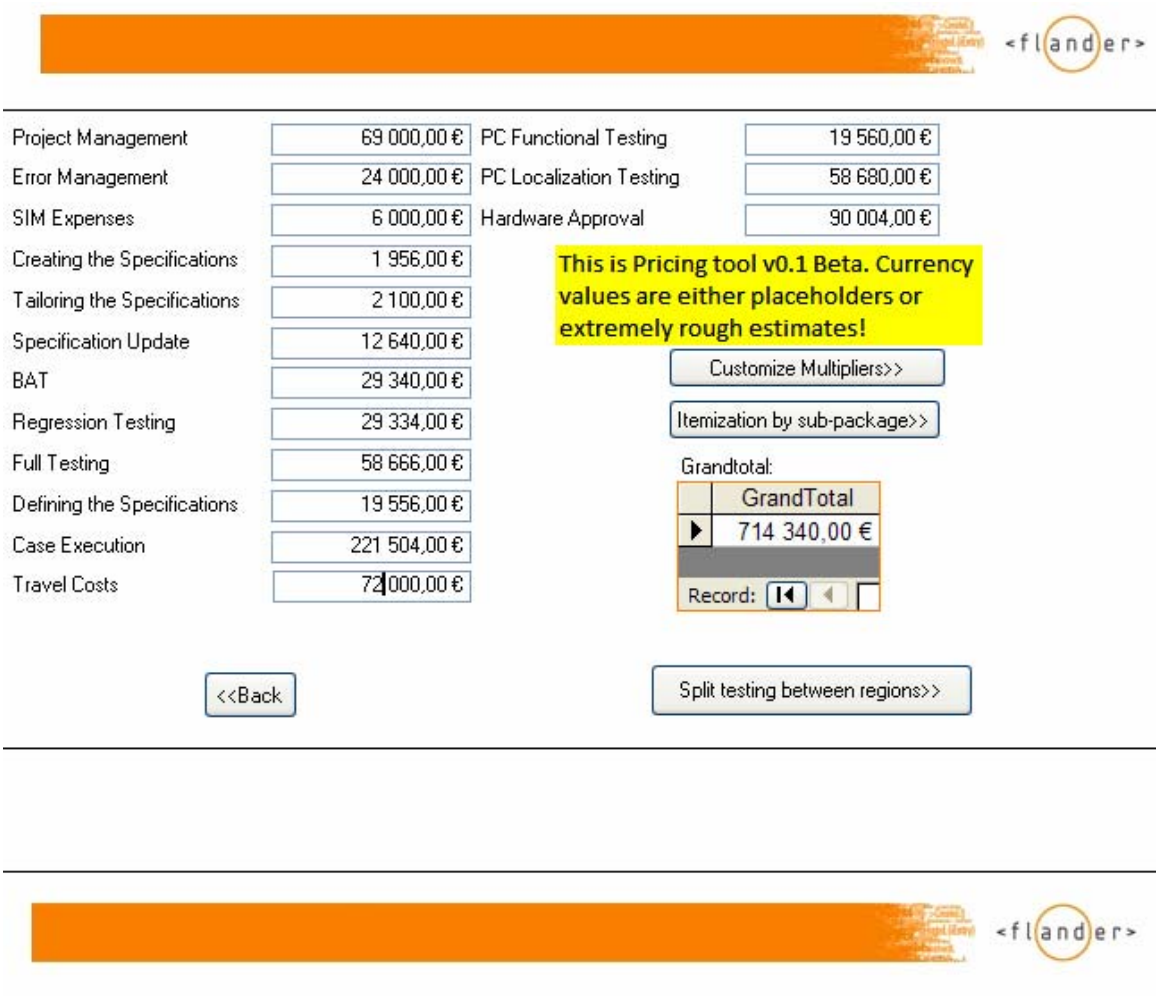


Figure 5-6: Screen 5 - Pricing Itemization

Its main purpose is to serve as a checkpoint, allowing the user to see a detailed break-down of the pricing package selected thus far, in itemized form. This is a more in-depth break-down than the one which will be presented to the user at screen 5B

Screen 5B will focus purely on the testing-related costs whereas this screen specifies both testing and project management costs, in addition to specification and travel related costs.

Screen 5B is displayed at the conclusion of the process and is meant to serve as a sort of receipt of the entire transaction, whereas Screen 5 serves as a running update of the current pricing situation, allowing the user to identify, with greater accuracy, where the major costs are and intuitively allowing for further adjustments via the “Custom Multiplier” link.

The figures are not editable at this stage, but navigation on this page allows for access to and modification of the multipliers which affect the various pricing sub-components. Simple, direct access to these variables by authorized users allows for easy updating and adjustment of the

figures on a per case basis. The multipliers can also be reset to their default values with the click of a button

5.1.6 Screen 5A: Custom Multipliers

The database was conceived based on the ER (Entity-Relationship) model which is the most common relational-database form. It was chosen because its structure enables rapid and efficient transformation of abstract data into more “concrete”, manageable and easily organized units.

The ER model describes the data involved in a real-world enterprise in terms of objects and their relationships and is widely used to develop an initial database design. Additionally, it provides useful concepts that allow us to move from an informal description of what users want from their database to a more detailed, and precise, description that can be implemented in a DBMS. (Ramakrishnan&Gehrke 2005)

Following the ER model logic, the sixth screen allows for fine-tuning pricing packages on a per-case basis. This is accomplished by enabling authorized users to modify specific multipliers (attributes) used in calculating the cost of each aspect of testing.

Refer to the screen-shot. Figure 5-7, below for a clearer perspective. The numbers in the boxes are attributes of the Package_ID entity, representing the amount of testing rounds for each category of testing.

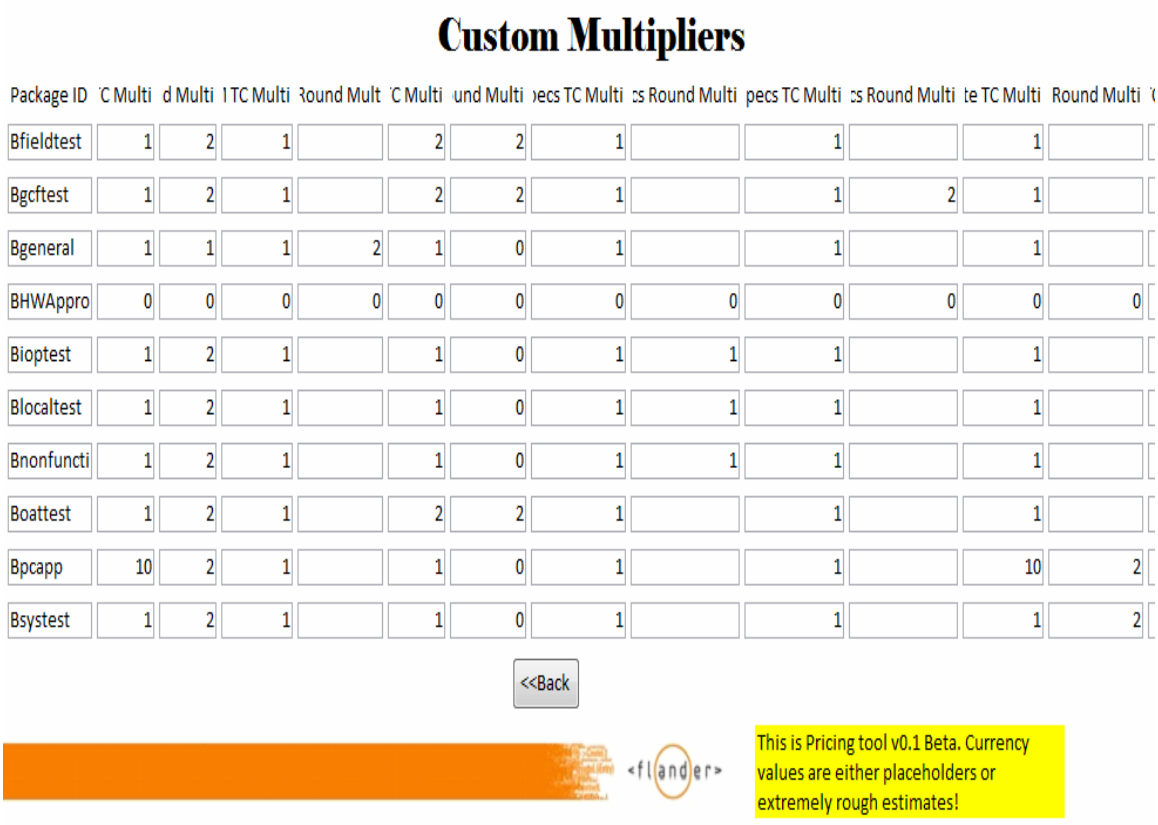


Figure 5-7: Screen 5A - Custom Multipliers

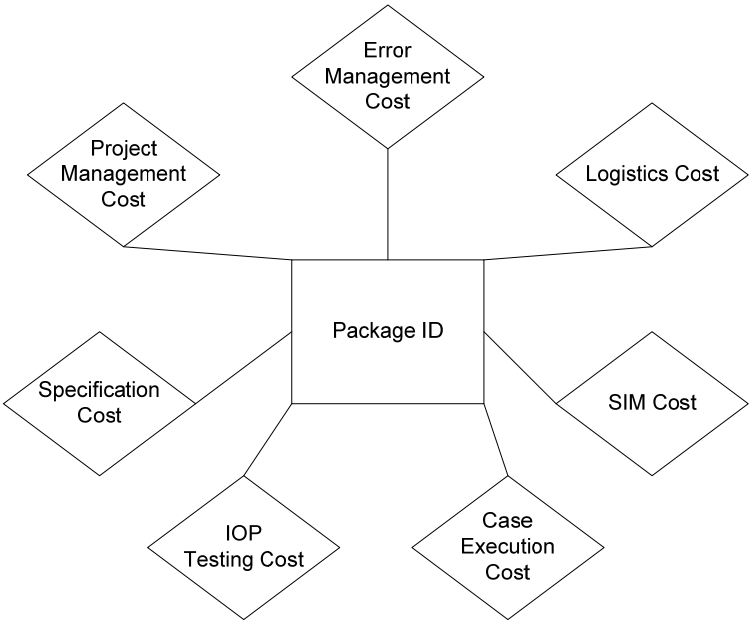


Figure 5-8: The *Package ID* Entity Set Displaying a Partial Selection of Attributes

Figure 5-8 above depicts the ER model manifestation of this database in rudimentary form.

The modifiable multipliers on this screen are part of an entity set called *Package_ID* residing in the table *Testing_Breakdown_All.* The Package ID entity has 72 attributes in total, any of which can be modified by authorized users.

One of the 3 branches stemming from Screen 5 terminates here. Once any multiplier adjustments have been applied, navigation within this screen is limited to back-tracking to the previous screen.

Once the user returns to the junction point represented by screen 5, he is presented with the option to navigate to Screen 5B: Itemization by Sub-package.

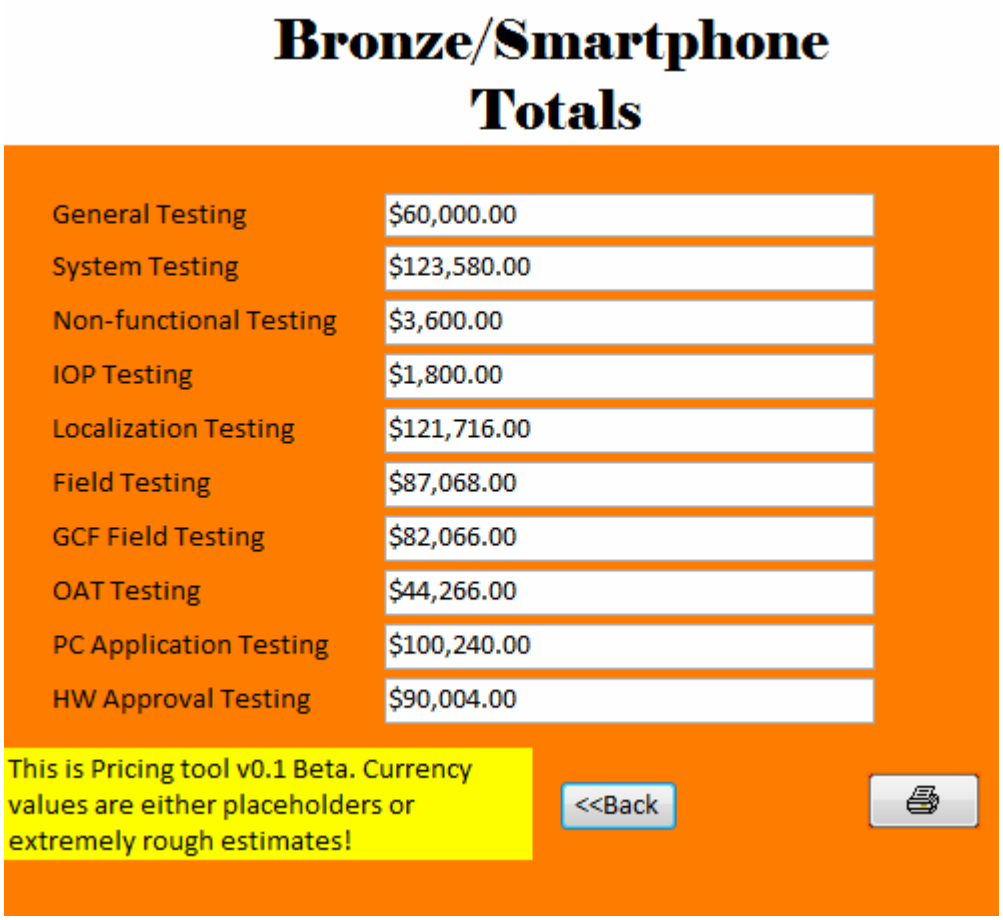
5.1.7 Screen 5B: Itemization by Sub-package

Screen 7 is purely a reference screen, similar in nature to Screen 5. Its data is derived from the variables chosen in Screen 2.

If the correct combination of variables were chosen in Screen 2, the user would have been presented with a pricing package recommendation rather than a blank query.

Screen 7 analyzes that recommendation and condenses the results into an itemized list of costs per major testing category, allowing the user to view and/or print out these results.

Refer to the screenshot below:



Figure… 5-9: Screen 5B - Itemization by Sub-package

The main objectives of Screen 7 are to:

1. Provide the user with an optimized view of the pricing package costs by combining related costs, thereby defragmenting the results and condensing them into broader cost categories
2. Provide the user with the option to create a hardcopy of these results for record-keeping purposes

Screen 5B represents the second termination point originating from Screen 5. In order to progress to 5C, and final screen, the user must back-track to Screen 5 and click on the “Split Testing Between Regions” button.

5.1.8 Screen 5C: Splitting Testing Between Regions

Screen 8 becomes relevant if, as stated earlier, the user has chosen to divide some portion of the testing workload between regions, namely Finland and China.

As noted earlier, Flander has extended its operations to China. This was accomplished by their acquisition of the Chinese-based Hexin Group. According to Flander CEO Markus Suomi, this merger enables Flander to take on larger-scale testing projects. (Flander 2007)

This growth affords Flander cost reductions through economy of scale. Screen 5C accounts for these cost reductions by enabling individual testing modules to be relegated to China with the applicable discount(s).

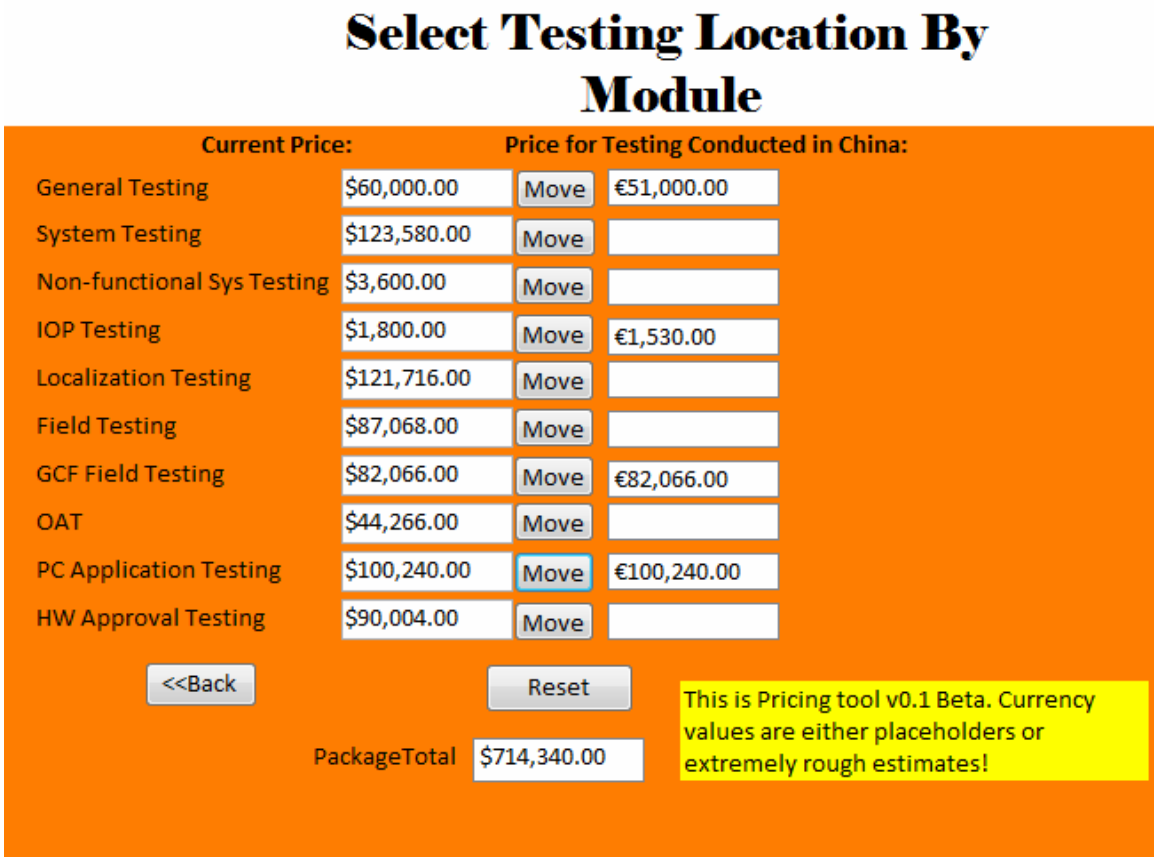


Figure 5-10: Screen 5C - Splitting Testing Between Regions

Each “Move” button utilizes an embedded query to derive its results. Unlike previous queries contained in this database, the source query for this screen is static. The code behind the buttons looks like this:

Option Compare Database

Private Sub Command56_Click()
systestAsian.RowSource = "SELECT qryAsianBronzeBySubPackage.SystemTesting FROM qryAsianBronzeBySubPackage; "
End Sub

Private Sub Command79_Click()
Loc.RowSource = "SELECT qryAsianBronzeBySubPackage.Loc FROM qryAsianBronzeBySubPackage; "
End Sub

Private Sub Command80_Click()
Fiel.RowSource = "SELECT qryAsianBronzeBySubPackage.FieldTest FROM qryAsianBronzeBySubPackage; "
End Sub

Private Sub Command81_Click()
gcf.RowSource = "SELECT qryAsianBronzeBySubPackage.GCF_FieldTotal FROM qryAsianBronzeBySubPackage; "
End Sub

Private Sub Command82_Click()
oat.RowSource = "SELECT qryAsianBronzeBySubPackage.OatTestTotal FROM qryAsianBronzeBySubPackage; "
End Sub

Private Sub Command83_Click()
pc.RowSource = "SELECT qryAsianBronzeBySubPackage.PCApptot FROM qryAsianBronzeBySubPackage; "
End Sub

Private Sub Command84_Click()
hw.RowSource = "SELECT qryAsianBronzeBySubPackage.Expr1 FROM qryAsianBronzeBySubPackage; "
End Sub

Private Sub Reset_Click()
systestAsian.RowSource = "Null"
Gen.RowSource = "Null"
lstNonf.RowSource = "Null"
iop.RowSource = "Null"
Loc.RowSource = "Null"
Fiel.RowSource = "Null"

```
gcf.RowSource = “Null”
oat.RowSource = “Null”
pc.RowSource = “Null”
hw.RowSource = “Null”
End Sub

Private Sub Command67_Click()
Gen.RowSource = “SELECT qryAsianBronzeBySubPackage.General FROM
qryAsianBronzeBySubPackage; “
End Sub

Private Sub Command77_Click()
lstNonf.RowSource = “SELECT qryAsianBronzeBySubPackage.Nonfunc FROM
qryAsianBronzeBySubPackage; “
End Sub

Private Sub Command78_Click()
iop.RowSource = “SELECT qryAsianBronzeBySubPackage.IOP FROM
qryAsianBronzeBySubPackage; “
End Sub
Private Sub Command87_Click()
On Error GoTo Err_Command87_Click

Dim stDocName As String
Dim stLinkCriteria As String

stDocName = “frm3Bronze_Subtotals”
DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command87_Click:
Exit Sub

Err_Command87_Click:
MsgBox Err.Description
Resume Exit_Command87_Click

End Sub
```

The code performs the following actions:

1. Specifies the query from which to retrieve the data. In this case, it’s *qryAsianBronzeBySubpackage*
2. For each click of the “Move” button, it runs a sub-query on the aforementioned static query, then auto refreshes
3. Clicking the “Reset” button executes a separate set of code which resets all fields to Null.

6 Pricing Tool Deployment

The pricing tool was set to be deployed at the 3GSM Convention in Barcelona in February 2006. This objective was never achieved and the project failure may be attributed to several factors. These include:

1. Poor database requirements analysis
2. Poorly defined/nonexistent project deliverables
3. An overhaul in management structure at Flander during the course of the project
4. Monitoring and controlling deficiencies

6.1 Reasons for Deployment Failure and Possible Remedies

The first step in evaluating the situation was to identify the possible causes for implementation failure. In an effort to extract valuable feedback from deployment failure, once identified, the factors attributing to this outcome require further analysis.

6.1.1Poor Database Requirements Analysis

Ramakrishnan&Gehrke specify certain critical elements to consider when planning a database. These aspects include:

- Understanding what data is to be stored
- What operations are most frequent and subject to performance requirements
- A formal methodology should be adopted for organizing and presenting information gathered in this step

The project was weakened from the start as 2 of these critical areas were not properly addressed.

The issue of frequent operations and performance requirements was never taken into consideration. Had this been addressed initially, it would have been determined that dynamic queries could have smoothly handled all the frequent operations, thus optimizing performance.

This oversight greatly increased product development time because performance bottlenecks were not actively sought out and addressed, but rather stumbled upon randomly.

The other critical factor which was ignored was the adoption of a formal methodology for information handling. Information was gathered through one-on-one and group meetings, (many of which were impromptu) but not always documented accurately because of time constraints or general lack of preparedness on part of the note-taker.

A more thought-out and methodical system of information gathering would have increased the efficiency of product development.

6.1.2 Poorly Defined/Nonexistent Deliverables

According to the Guide to the Project Management Body of Knowledge, a key element of any project's lifecycle is describing deliverables, defining when the deliverables are to be generated in each phase and how each deliverable is reviewed, verified, and validated.

A joint UK study, conducted by the NAO/OGC (National Audit Office/Office of Government Commerce) confirms that one of the major sources of project failure are inadequately defined Critical Success Factors (CSF's). On a short-term, small-scale project such as the development of the pricing tool, phase-specific deliverables can be considered CSF's.

In many instances during this project there were no clear cut deliverables defined. This made the process of creating and adhering to a schedule, as well as tracking progress, difficult.

A stricter system of managing deliverables would have enabled accurate schedules to be drawn up which would have kept the project more focused and on track and would have generated valuable feedback critical to the success of the project.

6.1.3 Management Structure Overhaul

Flander underwent an unforeseen major overhaul in upper-management during the course of the project. The upheaval resulted in one of the key project players and proponents of the Pricing tool, Mika Heikinheimo, being transferred from Tampere to Beijing.

Mr. Heikinheimo was not only the founder of Flander, but had also been heavily involved with the project from its conception. His deep organizational knowledge and coupled with his industry experience and support for the tool's development provided, not only a powerful engine for driving progress, but also a strong guiding force, ensuring that the project was moving along the right lines. His transfer left a vacuum, which severely hindered the progress of the pricing tool.

This loss of various levels of knowledge through, retirement, personnel shifts, etc. is referred to as *brain-drain* and not only affects the organization on a project level, but also has far-reaching negative implications. In addition to the loss of expertise and on the job knowledge built up over employees' careers, the loss of client intelligence, established internal and external networks and loss of social and networking skills may also reduce organizational performance. (Poole & Sheehan)

Poole & Sheehan describe the method for combating brain drain as a 3 step process:

1. **Recognizing key knowledge areas:** Analyzing information to determine what is critical to maintaining market competitiveness.
2. **Recognizing key knowledge holders:** Perform regular audits to assess the skills and specialized knowledge attributes of personnel organization-wide.
3. **Application of certain strategies to prevent knowledge loss:** Retaining a culture of sharing on an organizational level helps foster efficient knowledge transfer between coworkers by encouraging person-to-person interaction. Implementing COP's, *communities of practice*, (groups of globally-dispersed, like-minded people, sharing similar job descriptions) assists the knowledge transfer process by pooling knowledge.

Data gathered from points 1 and 2 above can then be plotted on a graph known as a *Knowledge Map*, enabling an organization to assess critical areas of knowledge drain and accurately gauge where their strengths and weakness lie, allowing problematic areas to be specifically addressed. See Figure 6-1 below:

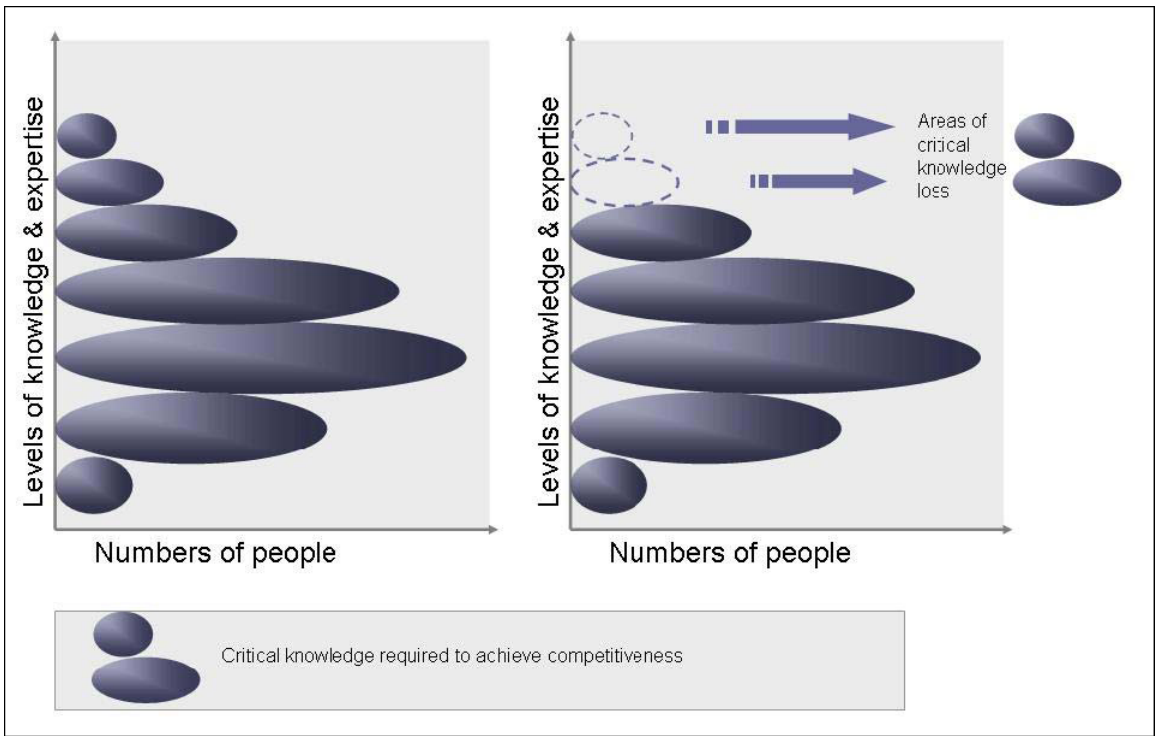


Figure 6-1: Example of a Knowledge Map
(*Source Strategies for Managing the Brain Drain – Implications for Knowledge Management*)

6.1.4 Monitoring and Controlling Deficiencies

The biggest flaw in the design of this project plan was the failure to incorporate a formal system for monitoring and controlling output.

An insufficient amount of control mechanisms, inefficient controlling techniques and lapses in the controlling process all serve to obscure the true status of a project from its team members. This obscurity is one of the major causes of project failure because corrective action cannot be taken if key players are unaware that a problem exists. (McConnell 1996)

Output monitoring was accomplished informally and inconsistently by randomly reviewing the project status and briefly comparing its state to the desired state of the finished product. This method provided no true metric and the results were wholly subjective as opposed to objective.

Had such a methodology been adopted, it would have mitigated the damages caused by the inadequately defined deliverables as well as the unforeseen management restructuring.

Efficient monitoring and controlling would have identified those aforementioned deficiencies early, allowing for adequate time to address and correct the issues and potentially salvage the project.

7 Conclusion

Any project, regardless of scale, requires thorough planning, a methodical system of monitoring and controlling activities and a means of measuring output in order improve the likelihood of successful completion.

Each phase demands careful execution, open communication and a concerted effort amongst project team members in order to generate the specified deliverables in timely fashion.

Even on the most carefully orchestrated project, a certain amount of deviation from the guidelines laid out in the project plan is inevitable. Unforeseen and usually uncontrollable circumstances such as budgeting issues, deadline changes or shifts in personnel are bound to arise even in the most ideal conditions.

Because even successful projects may have problems which are often unavoidable, project failure results not necessarily from the nature of the issue or issues, but from failure to identify them in a timely manner, thus not allowing for the pursuit of effective corrective action.

As was the case with the pricing tool, a few missteps in critical areas can ultimately doom a project, regardless of the optimism of the parties involved or the amount of resources expended.

Project risk can never be completely eliminated only mitigated. Therefore a project's success can never be guaranteed regardless of how meticulously planned or how well-executed its phases.

What is certain, however is that poorly planned and executed projects run a higher risk of failure and when failure occurs, dissection of the case will ultimately provide clues as to why this outcome was rendered. Through proper analysis of these failure factors, one gains insight into how they may be avoided in future projects

In dissecting the pricing tool case, the evidence supports the claim that a haphazard system of the monitoring and controlling enabled otherwise manageable deficiencies to remain unchecked, thus allowing them to spiral out of control until their force was powerful enough to derail the entire project.

Although the implementation objective was not achieved, valuable feedback may be extracted by thorough analysis of the factors attributing to the lack of attainment of the goal.

Project team members were ultimately enriched by the insight gained during the process of determining how to avoid these same pitfalls in future projects.

8 References

1. Atzeni, Paolo et. al. Database Systems – Concepts, Languages and Architectures. New York: McGraw Hill, 1999
2. Bouma, Frans. Stored Procedures vs. Dynamic Queries, 2007, Blog excerpt: http://weblogs.asp.net/fbouma/archive/2003/05/14/7008.aspx
3. Flander Press Release. *Flander Acquires Hexin's Mobile Testing Business*. June 20, 2007
4. Lindholm Christian et. al. Mobile Usability: How Nokia Changed the Face of the Mobile Phone. McGraw Hill, 2003
5. McConnell, Steven. Rapid Development: Taming Wild Software Schedules. Redmond: Microsoft Press, 1996
6. NAO/OGC. *Common Causes for Project Failure*, 2003, Archived article: http://www.ogc.gov.uk/documents/Project_Failure.pdf
7. Pagani, Margherita. Mobile and Wireless Systems: Beyond 3G. Hershey: The IRM Press, 2005
8. Patri, Rahul. *Testing Communications Between Machines.* Software Testing and Performance Magazine, April 2006
9. Poole&Sheehan. *Strategies for Managing the Brain Drain – Implications forKnowledge Management.* Arup Management
10. Project Management Institute. A Guide to the Project Management Body of Knowledge: PMBOK Guide 3rd ed. Project Management Institute Inc: 2004
11. Ramakrishnan, Ragu and Gehrke, Johannes. Database Management Systems. The McGraw Company, 2001
12. Speelpenning, Jan et. al. Oracle – Data Modeling and Relational Database Design. Oracle Corporation, 2001
13. Wood, David. Principles of Successful Smartphone Development Projects John Wiley&Sons Lt,: 2005

Appendices

Appendix A

Appendix A: Pricing Tool Baseline Figures

Gold Pricing Package

Silver Pricing Package

Bronze Pricing Package

Appendix B

Initial Pricing Tool Proposal

1.Select phone type:

- [ ] Basic
- [ ] Feature
- [ ] Smartphone

2.Select Feature/Smartphone features:

- [ ] WCDMA (3G)
- [ ] Browser
- [ ] Java
- [ ] WinCE
- [ ] GPS

Input Methods:

- [ ] Touchscreen
- [ ] Qwerty Keyboard

Connectivity

- [ ] USB
- [ ] IR
- [ ] Bluetooth
- [ ] WLAN
- [ ] PTT

Multimedia

- [ ] Camera
- [ ] MP3 Player
- [ ] Video Player
- [ ] Radio

Messaging

- [ ] MMS
- [ ] E-mail
- [ ] FAX /Printing capabilities

Bundled Software

- [ ] Office Applications
- [ ] Connectivity packages (PC Suite)

3.Select languages to be tested:

- [ ] US English
- [ ] British English
- [ ] European French
- [ ] Canadian French
- [ ] Italian
- [ ] German

- [ ] European Spanish
- [ ] Mexican Spanish
- [ ] Brazilian Portuguese
- [ ] European Portuguese
- [ ] Danish
- [ ] Dutch
- [ ] Norwegian
- [ ] Swedish
- [ ] Finnish
- [ ] Russian

4.Select Testing Package:

- [ ] Bronze
- [ ] Silver

Desired number of system testing rounds [ ] Desired number of localization testing rounds [ ] Field Testing ( )

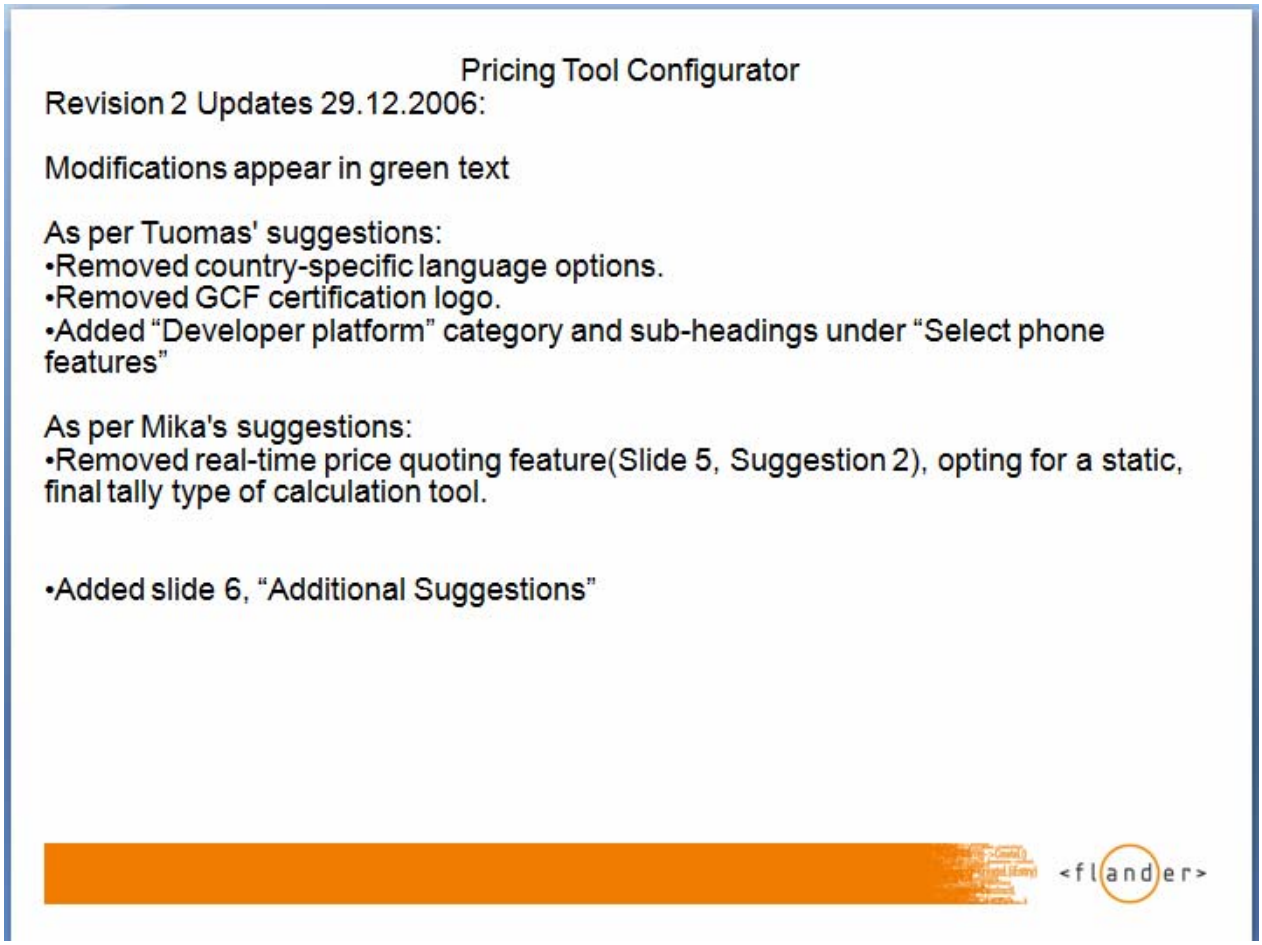
- [ ] Gold

Desired number of system testing rounds [ ] Desired number of localization testing rounds [ ] Field Testing ( )

Submit

Appendix C

Final Pricing Tool Proposal



Preliminary Pricing Configurator

Screen **1**:

Select Phone Type

Description: A mutually-exclusive list of choices, each containing a thumbnail image and brief description of the phone type to aid in making an accurate selection.

-**Basic**: Low-end, bare-bones, minimalistic models with a modest amount of features; usually not much more than basic call, SMS and polyphonic ringtone capabilities.

-**Feature:** Mid-range models with a wide-array of features, usually utilizing 1 or more standout features such as a megapixel camera, media player or GPS capabilities. Powered by a bona-fide operating system such as Symbian.

-**Smartphone:** Upper-tier devices incorporating aspects of both a PDA (touchscreen, qwerty keyboard) and feature phone (Megapixel camera) into one comprehensive package

Screen **2**:

Select device features

- Checklist covering the defining characteristics of the device.

- A "Developer Platform" heading containing sub-headings such as: "Symbian", "Java (CDC, CLDC etc.)"... should be included for manufacturers' benchmarking purposes.
- The Pricing tool should be capable of processing the selected features, generating a profile, then comparing it with predetermined criteria from screen 1 to **ensure consistency** between the "phone type" selection and device's actual features.

Example: User selects "Feature" phone in screen 1, but checks off options which are consistent with a smartphone in screen 2. User is notified of this discrepency and offered suggestions on how to address it.

Screen **3**:

Region/Country/Language Selection

- User is guided through a 3 step language selection process:
  1. User is presented with a "Region" selection menu, i.e. Europe, Asia-Pacific, the Americas.
  2. After region selection, a list of region-specific countries is displayed.
  3. Once the country selection is processed, a list of languages is presented.

Note: Multiple selections should be supported throughout the entire process.

.

Screen **4**:

Testing Package Selection

•User is presented with the 3 testing package options, along with a general overview of each. The overview sections will briefly cover level-specific details such as: testing areas, number of rounds and the confidence level achieved upon completion. Individual level descriptions will also contain links ("more...") which will expound on each level.

Examples:

GOLD

The most comprehensive testing package available.Thorough in both extent and depth, the gold package consists of 6000 test cases, spanning 15 test rounds, ensuring a 99.9% confidence level...more

Suggestion: Pricing tool could harnesses user input from screens 1-3 to generate a testing package suggestion.

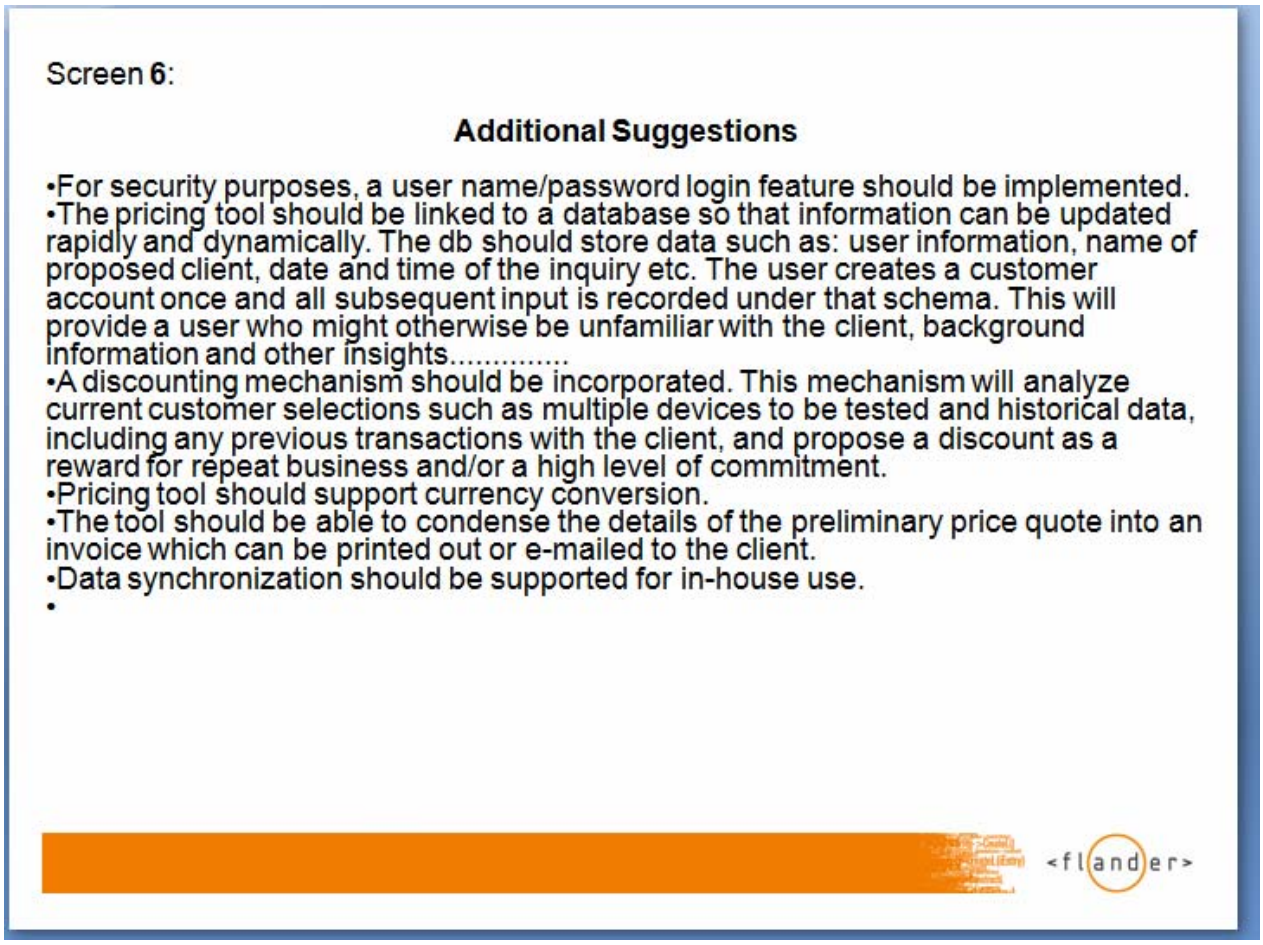
Screen 5:

Testing Package Details from the "more..." Link

•User was presented with screen 4, to gain perspective on the broad differences between the testing levels.
•Clicking on the "more..." link directs the user to a detailed view, further explaining the intricacies of each level. This screen will offer additional guidance to the user in the selection of a testing level.
•Details could include test areas, break-downs of test rounds, certification explanations and other criteria useful in determining the most appropriate testing solution based on customer needs/requirements.

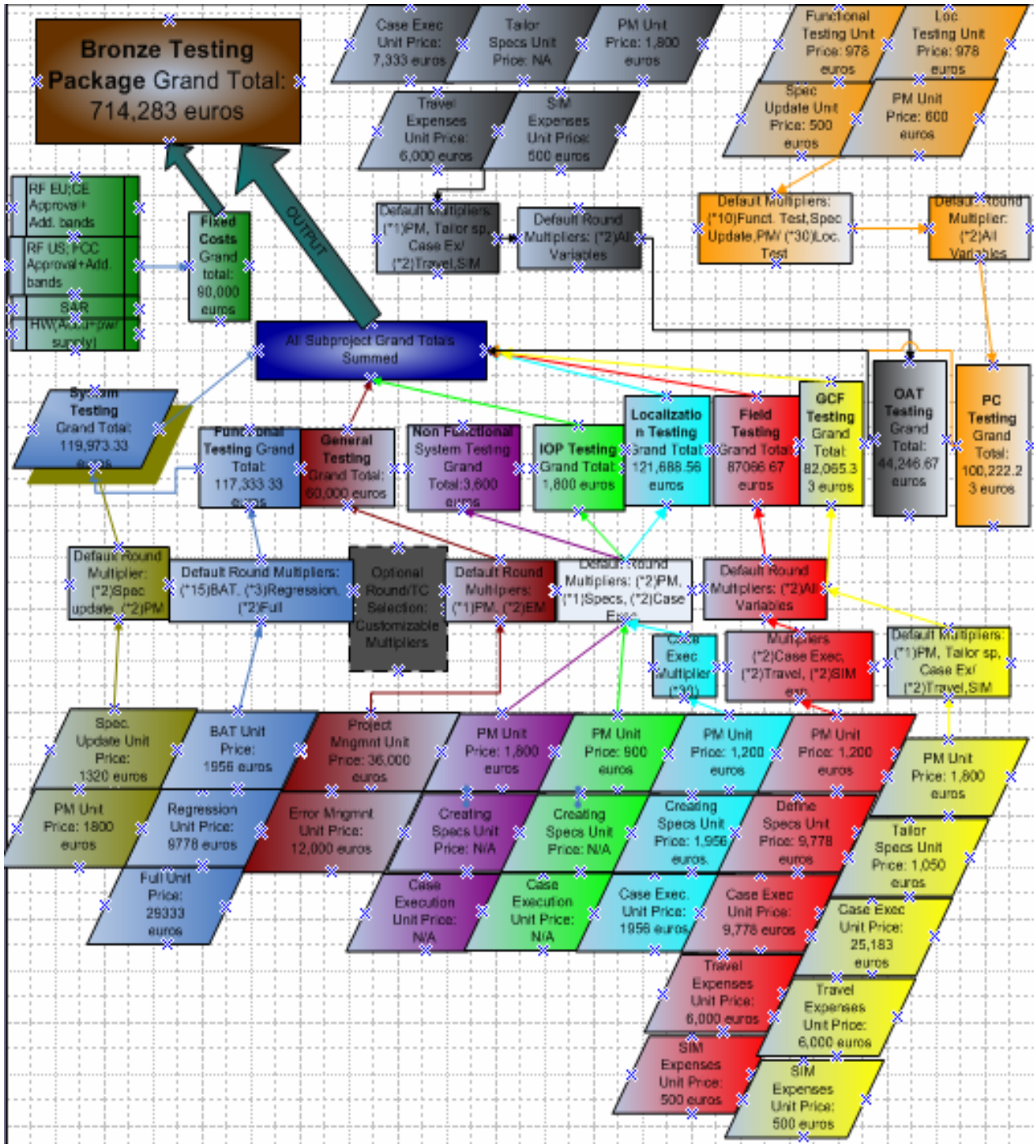
Suggestion 1: A certain degree of flexibility could be worked into the pricing model so that the user has the ability to tailor a package to suit their needs. There would be a tradeoff between the extent and depth of testing within the customization option. The pricing model will handle the mechanics so that the user can manually make a scale change in one area and the model will suggest an appropriate area in which an adjustment could be made to balance out the equation and retain the current price quote.

Suggestion 2: A calculate button could be incorporated to do the final tally at the end of the selection process.

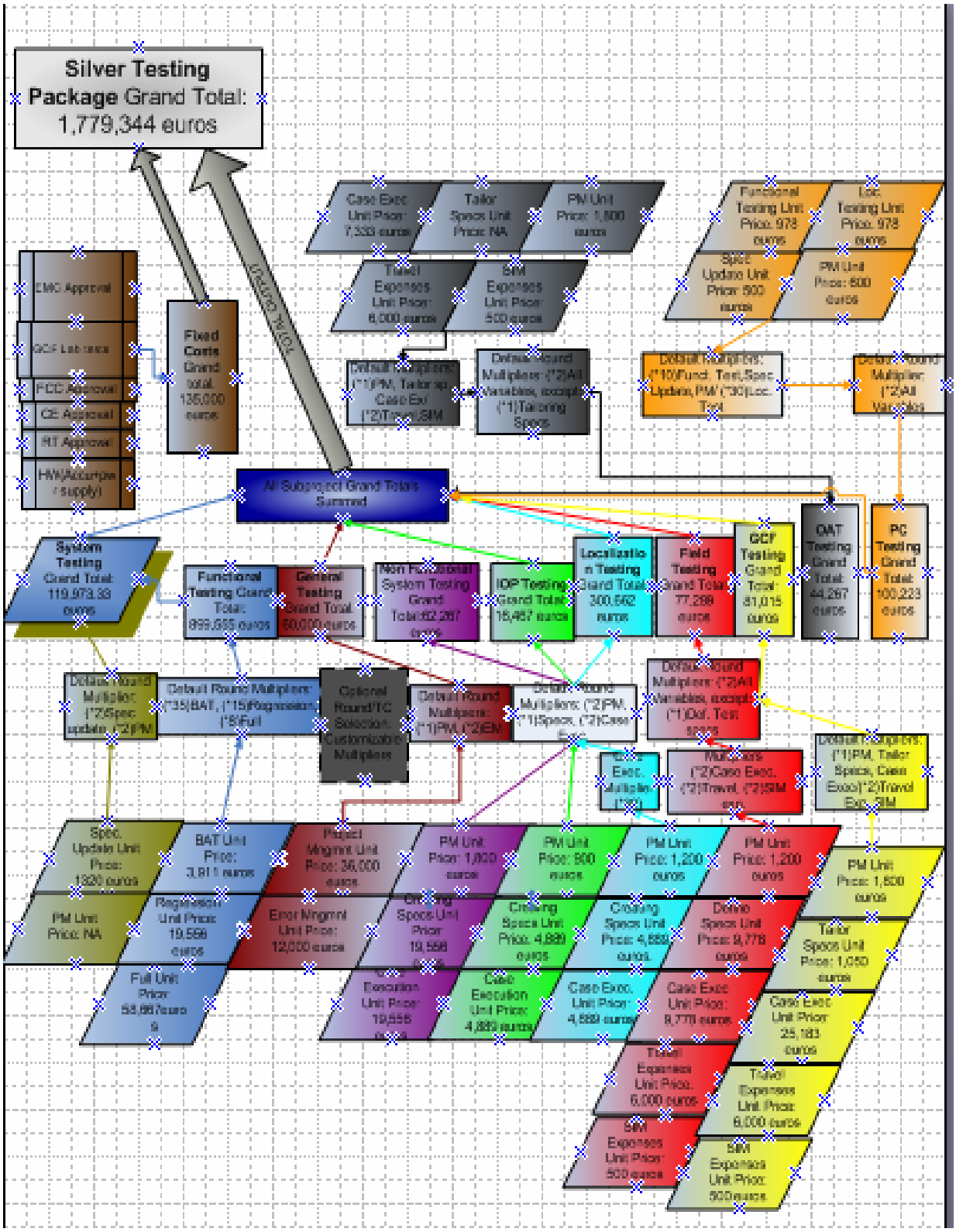


Appendix D:

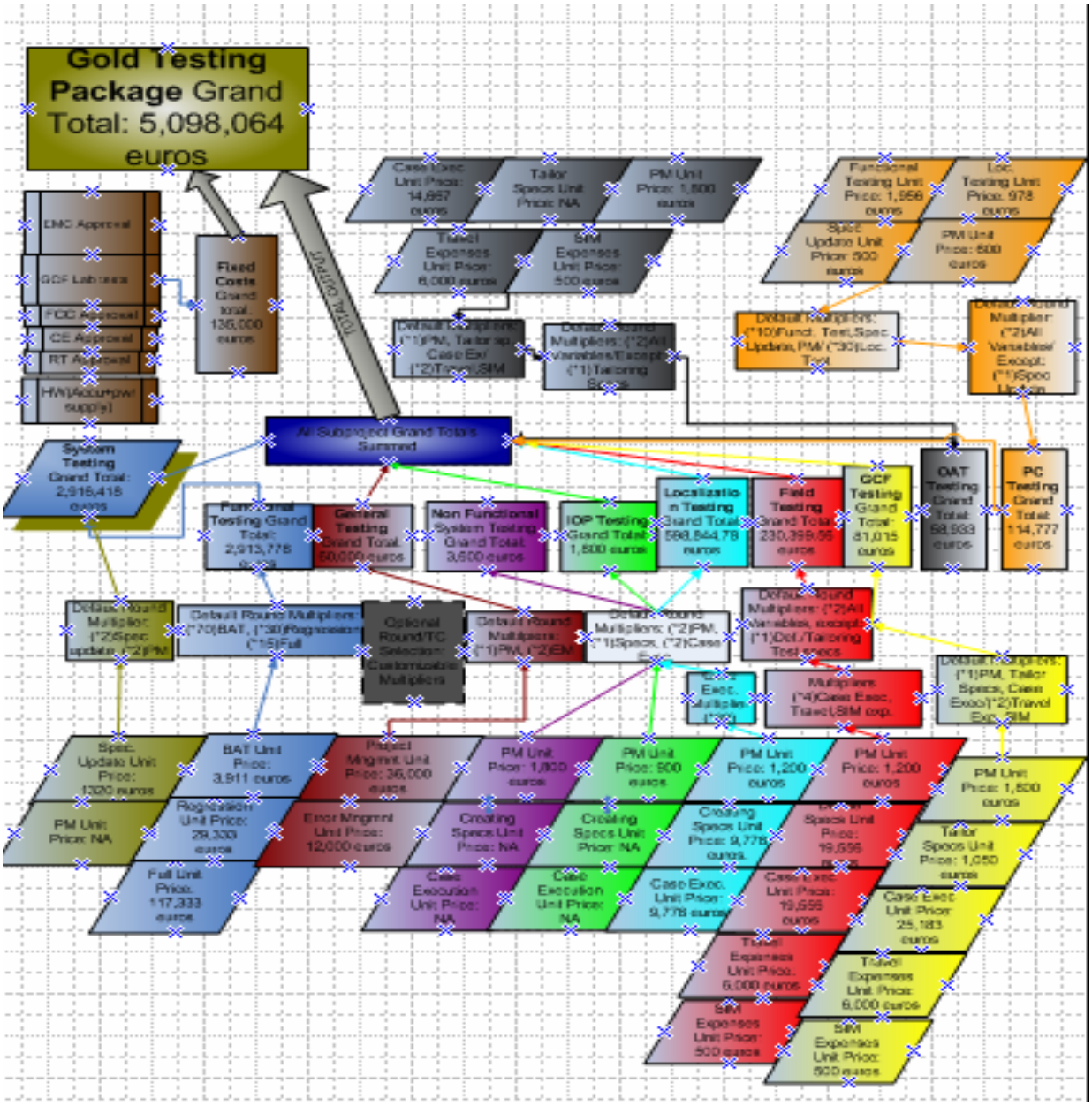
All Levels



Visual Depiction of the Bronze Pricing Package



Visual Depiction of the Silver Pricing Package



Visual Depiction of the Gold Pricing Package